all rights reversed

HACKER

Νοέμβριος 2014 # Τιμή τεύχους 3,99€

τεύχος 038 - The Note Edition



Low Orbit Ion Cannon VS iptables! [*]

Η τεχνική της στεγανογραφίας

Social engineering με το Maltego

Hacking με τα Android και Zimperium

Syncthing: Ο αντικαταστάτης του BtSync;

[*] Ανηλεείς επιθέσεις DoS που μας κάνουν καλό

all rights reversed

HACKER

Περιεχόμενα

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Γιώργος Αραβίδης, Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Πέτρος Κυλαδίτης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Θωμάς Σερμπίνης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το ηλεκτρονικό περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στην Καλαμαριά Θεσσαλονίκης. Ο δικτυακός τόπος του περιοδικού είναι στο http://deltahacker.gr. Το email επικοινωνίας είναι το talk2us@deltahacker.gr και ειδικά για τις συνδρομές είναι το subscriptions@deltahacker.gr. Οι γενικές πληροφορίες για τις συνδρομές παρατίθενται στο http://deltahacker.gr/subscriptions και οι ίδιες οι παραγγελίες γίνονται από το http://deltahacker.gr/pdf12order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

Σε όποιον αρέσει και τι να κάνουμε αν δεν μας αρέσει

Περίεργα επεισόδια είχαμε τις τελευταίες εβδομάδες στο Facebook. Για άλλη μια φορά, έστω και με κωμικό τρόπο, ήλθαν στο προσκήνιο ζητήματα που αφορούν στην προστασία της ιδιωτικότητας και των προσωπικών δεδομένων. Εδώ που τα λέμε, οι σχετικές συζητήσεις και οι προβληματισμοί είναι σχεδόν πάντα επίκαιροι. Μερικές φορές για καλό λόγο, κάποιες άλλες φορές εντελώς αδικαιολόγητα.

by subZraw

Εκείνο το post που κυκλοφόρησε πρόσφατα στο Facebook και κατάφερε να επιβιώσει για αρκετές μέρες, δύσκολα *δεν* θα το προσέξατε. Ήταν ένα κάπως παράξενο κείμενο, το οποίο με μια πολύ γρήγορη κι επιπόλαιη ματιά έμοιαζε να έχει νομική υπόσταση. Όποιος το αναρτούσε στο προφίλ του, υποτίθεται ότι αυτομάτως προστατευόταν από τις πρακτικές εκμετάλλευσης προσωπικών δεομένων από πλευράς Facebook. Επρόκειτο φυσικά για φάρσα, με την οποία όμως αρκετοί ξεγελάστηκαν και πράγματι δημοσίευσαν στα προφίλ τους το post-ασπίδα, κατά των πρακτικών του Facebook και υπέρ της διασφάλισης των προσωπικών τους δεδομένων.

Κοιτάξτε τώρα τι συμβαίνει – ή για την ακρίβεια τι νομίζουμε ότι συμβαίνει. Υπάρχουν, που λέτε, δύο κατηγορίες ανθρώπων: Εκείνοι που χωρίζουν τους ανθρώπους σε δύο κατηγορίες, κι εκείνοι που δεν χωρίζουν τους ανθρώπους σε δύο κατηγορίες. Μερικές φορές, στην προσπάθειά μας να βγάζουμε λίγο νόημα για πράγματα που συμβαίνουν γύρω μας, ομολογούμε ότι κάνουμε ό,τι κάνουν οι άνθρωποι της πρώτης κατηγορίας. Μπαίνουμε δηλαδή στον πειρασμό και, με αφορμή κάποιο συμβάν που έχει προκαλέσει αίσθηση, χωρίζουμε τους ανθρώπους σε δύο κατηγορίες. Και λέμε "μπαίνουμε στον πειρασμό", αφού πολλά από όσα συμβαίνουν είναι λίγο περισσότερο περίπλοκα απ' όσο αρχικά δείχνουν, ώστε να δικαιούμαστε να συζητάμε με αφετηρία δύο μόνο κατηγορίες. Αλλά για να μη γενικολογούμε άλλο, ας επιστρέψουμε στο περιστατικό με την πρόσφατη φάρσα που έλαβε χώρα στο Facebook. Παρατηρώντας, λοιπόν, τις αντδράσεις των χρηστών, ισχυριζόμαστε ότι όσοι έχουν λογαριασμό στο Facebook χωρίζονται σε δύο κατηγορίες:

- **κατηγορία 1:** "μπαναΐα 'μ, καλά που το 'δα – πανατοποστάρω ή/και να κάνω λάικ αμέσως, αφού προφανώς και με συμφέρει/βοηθά/κάνει καλό/είναι ευλογία"
- **κατηγορία 2:** "ρε τι χαϊβάνια 'στε 'σεις, που πάτε και ποστάρετε ή/και κάνετε λάικ σε όποια π@π@ρ1@ δείτε – χαχαχα!"

Αμέσως αμέσως φαίνονται οι ατέλειες της διπλής κατηγοριοποίησης, αφού για το χώρο που εξετάζουμε πανεύκολα διακρίνουμε και μια τρίτη κατηγορία: Σ' αυτήν ανήκουν όλοι όσοι έχουν λογαριασμό στο Facebook, όμως την ίδια στιγμή δεν παίρνουν τόσο πολύ σοβαρά τη δική τους παρουσία εκεί, ούτε την παρουσία των άλλων, οπότε και δεν σπεύδουν να κάνουν αναδημοσιεύσεις ή λάικ, αλλά ούτε και να σχολιάζουν τους άλλους που σπεύδουν. Όμως για λόγους που βοηθούν τη συζήτησή μας και την οδηγούν προς τα εκεί που θέλουμε να καταλήξουμε, ας αφήσουμε κατά μέρος αυτή την 3η κατηγορία – ή όποια άλλη κατηγορία μπορείτε να σκεφτείτε, τέλος πάντων.

Το Facebook, αγαπητές κι αγαπητοί, μοιάζει να είναι μια δωρεάν υπηρεσία. Σίγουρα συμπεριφέρεται ως μία τέτοια, αφού κατά την εγγραφή αλλά κι αργότερα, κατά τη χρήση, δεν μας ζητείται να πληρώσουμε. Εντάξει, αν θέλουμε ν' ανεβάσουμε μια διαφήμιση,

τότε καθ' όλη τη διάρκεια της προβολής της πληρώνουμε, όμως αυτό είναι λογικό να συμβαίνει κι όλοι καταλαβαίνουν γιατί. Στην πραγματικότητα, όμως, πολλά πράγματα που μοιάζουν να είναι δωρεάν, ε, δεν είναι. Και δεν είναι δωρεάν είτε επειδή είναι αδύνατον να είναι δωρεάν, είτε επειδή από σχεδίαση δεν είναι. Το Facebook συγκαταλέγεται σ' αυτά τα πράγματα: Δείχνει να είναι δωρεάν, όμως στην πραγματικότητα όλοι όσοι το χρησιμοποιούμε, πληρώνουμε. Όχι με χρήματα, αλλά καταθέτοντας δεδομένα που μας αφορούν και κάνοντας ενέργειες που κάτι λένε για 'μας.

Ένα τεράστιο κοινωνικό δίκτυο σαν το Facebook, είναι απλά αδύνατον να λειτουργεί δωρεάν. Το ίδιο ισχύει και για πολύ μικρότερα δίκτυα ή υπηρεσίες, όμως ας παραμείνουμε στο Facebook. Το Facebook είναι μια επιχείρηση που έχει **α)** τρομερά μεγάλα λειτουργικά έξοδα και **β)** δικαίωμα στο κέρδος. Τα έξοδα του Facebook δεν καλύπτονται από καλοκάγαθους μεγιστάνες που όλη την ώρα ψάχνουν τρόπους για να μοιράζουν τα χρήματά τους. Τα όποια χρήματα δίνει ένας οποιοσδήποτε επενδυτής περιμένει να τα πάρει πίσω με κέρδος. Έτσι λειτουργεί ο κόσμος μας. Μια οποιαδήποτε εταιρεία που δεν είναι μη-κερδοσκοπική και δεν έχει τρόπο να αποκτά (διαρκώς) δωρεάν χρήματα, προκειμένου να επιβιώνει είναι υποχρεωμένη να επιζητά το κέρδος. Ξανά: Έτσι λειτουργεί ο κόσμος μας. Το Facebook, ως μια εταιρεία αυτού του κόσμου, προσπαθεί να κερδίζει με δύο τρόπους: Με το να χρεώνει όποιον θέλει να διαφημίζεται στην πλατφόρμα της, αλλά κυρίως με το να πουλά δεδομένα μας σε άλλους, οι οποίοι διαφημίζουν στο Facebook και ψάχνουν για πελάτες ανάμεσά μας. Να πώς πληρώνουμε *όλοι* στο Facebook: Όχι κατ' ανάγκη με χρήματα, αλλά με δεδομένα που μας αφορούν.

Στους όρους χρήσης του Facebook, αυτούς που κανείς μας δεν διαβάζει αλλά όλοι αποδεχόμαστε, αναφέρονται μεταξύ άλλων και τα δικαιώματα της εταιρείας όσον αφορά στα δεδομένα μας. Το να έχει αποδεχθεί κάποιος αυτούς τους ίδιους όρους και μετά να πιστεύει ότι με μια απλή ανάρτηση στο προφίλ του θα απαγορεύει από το Facebook να κερδίζει από εκείνον, είναι τουλάχιστον αφελές. Αν, καλέ μου άνθρωπε, δεν σου αρέσουν οι όροι χρήσης του Facebook, σου έχουμε καλά νέα: Έχεις το ελεύθερο να μην τους αποδεχτείς και να μη φτιάξεις λογαριασμό στην υπηρεσία. Επίσης: Αν είσαι φυσιολογικός άνθρωπος και δεν διαβάζεις τους όρους χρήσης, έστω κι εκ των υστέρων έχεις το ελεύθερο να κλείσεις το λογαριασμό σου στην υπηρεσία. Δεν είναι κακό αυτό. Η αλλαγή γνώμης είναι δικαίωμα και ανά πάσα στιγμή θεωρήσεις ότι δεν σε χωράει άλλο ο τόπος εκεί μέσα, σήκω και φύγε. Αλλά ξέρετε κάτι; Υπάρχουν, ευτυχώς, κι άλλες εναλλακτικές.

Μπορεί, π.χ., να μην αποδέχεσαι πλήρως τους όρους χρήσης του Facebook, και παράλληλα να είσαι στην πλατφόρμα και να συνεχίσεις να τη χρησιμοποιείς. Δεν είμαστε δικηγόροι, σκεφτόμαστε όμως ότι μάλλον κανείς δεν πρόκειται να σε μηνύσει γι' αυτό. Πάρε λοιπόν το ρίσκο, ζήσε στα άκρα – και βεβαίως YOLO, baby!

Πέρα απ' την πλάκα, υπάρχουν πολλοί λόγοι για να θέλει κάποιος να βρίσκεται στο Facebook. Η επικοινωνία, η διασκέδαση αλλά και η ενημέρωση, είναι οι τρεις που αμέσως έρχονται κατά νου. Δεν είναι οι μοναδικοί: Για παράδειγμα, πιθανώς να θέλεις να είσαι στο Facebook για επαγγελματικούς λόγους. Τόσος κόσμος είναι εκεί, οπότε να μια ευκαιρία να προωθήσεις τη δουλειά σου, να προβάλλεις την επιχείρησή σου, να κάνεις επαγγελματικές γνωριμίες κ.λπ. κ.λπ. Αν την ίδια στιγμή δεν σου αρέσουν οι όροι χρήσης του Facebook ή ανησυχείς για πιθανή διαρροή/εκμετάλλευση ευαίσθητων δεδομένων που σε αφορούν, έχουμε να προτείνουμε μια λύση.

Ανεξάρτητα από το πόσο επιμελώς έχεις παίξει με τις ρυθμίσεις ασφαλείας κι απορρήτου του Facebook, ανά πάσα στιγμή να θεωρείς πως ότι κάνεις εκεί γίνεται δημόσια κι όχι μόνο για έναν περιορισμένο, ελεγχόμενο κύκλο χρηστών. Σκέψου, π.χ., ότι ακόμη κι αν σήμερα καθορίζεις πως μία συγκεκριμένη φωτογραφία μπορούν να τη βλέπουν μόνο πέντε κοντινοί φίλοι, ένα bug αύριο είναι πιθανό να τη βγάλει σε δημόσια θέα. Μπορείς να είσαι σίγουρος ότι ποτέ δεν θα συμβεί αυτό; Ακούσαμε την απάντηση που μόλις έδω-

σες – και ούτε για μια στιγμή πιστέψαμε ότι μπορείς :) Ας το πάμε τώρα ακόμη πιο πέρα: Από τη στιγμή που ανεβάζεις κάτι στο Facebook, έχεις χάσει τον έλεγχό αυτού του "κάτι". Επιστρέφοντας στο παράδειγμα της φωτογραφίας, ας πούμε ότι την ανέβασες κι έγινε η σχετική πλάκα. Μετά αποφασίζεις ότι είναι καλύτερα να τη διαγράψεις – και τη διαγράφεις. Πώς είσαι σίγουρος ότι έχει διαγραφεί οριστικά κάθε *παρουσία* της φωτογραφίας από τους servers του Facebook; Το ότι εσύ δεν βλέπεις πια τη φωτογραφία στο προφίλ σου, δεν σημαίνει ότι δεν υπάρχει κάπου το αντίστοιχο αρχείο.

Νομίζουμε ότι η άποψή μας για το Facebook είναι πλέον ξεκάθαρη. Πρόκειται για άλλη μια εταιρεία, η οποία έχει δικαίωμα να επιδιώκει το κέρδος. Η δημιουργία λογαριασμού στο Facebook απαιτεί την αποδοχή συγκεκριμένων όρων χρήσης[1]. Αν δεν μας αρέσουν, δεν τους αποδεχόμαστε. Αν μας αρέσουν ή δεν πρόκειται να σκοτιστούμε διαβάζοντάς τους, τότε ας χρησιμοποιούμε το Facebook ως μια δημόσια, ουσιαστικά, πλατφόρμα: Ό,τι ανεβάζουμε εκεί, ό,τι γράφουμε, ό,τι σχολιάζουμε, όπου κάνουμε λάικ κ.λπ., καθόλου δεν αποκλείεται να φανεί κάποια στιγμή σε όλους, ασχέτως των ρυθμίσεων και των επιλογών μας. Ακόμα χειρότερα, πιθανώς να χρησιμοποιηθεί κι εναντίον μας. Ποτέ δεν μπορούμε να είμαστε 100% βέβαιοι.

Λίγη σύνεση, αγαπητές κι αγαπητοί, και θεωρούμε ότι με το Facebook δεν υπάρχει θέμα. Ενδέχεται όμως να υπάρξει θέμα με άλλες υπηρεσίες στο Internet, οι οποίες είναι πολύ πιο βασικές και κρίσιμες από μια πλατφόρμα κοινωνικής δικτύωσης. Θα έχετε ακούσει, για παράδειγμα, ότι ενώ η ηλεκτρονική αλληλογραφία κανονικά είναι μόνο για τα μάτια του αποστολέα και του παραλήπτη, στην πραγματικότητα έχουν πρόσβαση σ' αυτή και τρίτοι – αρκεί να το επιλέξουν κι ενδεχομένως να εφοδιαστούν με την κατάλληλη εισαγγελική εντολή. Πέρα από εισαγγελείς, πρόσβαση στην ηλεκτρονική μας αλληλογραφία είναι δυνατόν να έχει ακόμη και κάποιος διαχειριστής που απλά βαριέται. Είναι όμως τόσο εύκολο να συμβεί αυτό; Κοιτάξτε, από τη στιγμή που τα email *δεν* αποθηκεύονται κρυπτογραφημένα στους mail servers, ναι, είναι "τόσο εύκολο".

Για την περίπτωση του email, μπορούμε άραγε να πούμε κάτι παρόμοιο μ' αυτό που σίγουρα μπορούμε να πούμε για την περίπτωση του Facebook; Ισχύει κι εδώ το "έτσι είναι κι αν σ' αρέσει, αλλιώς φύγε"; Λέμε πως όχι, δεν ισχύει. Και δεν ισχύει, διότι το email είναι πολύ βασικό για να πάψουμε να το χρησιμοποιούμε επειδή κάτι *ίσως* συμβαίνει ή πρόκειται να συμβεί. Τι κάνουμε, λοιπόν; Μια πολύ καλή λύση, είναι η αποστολή email σε ισχυρά κρυπτογραφημένη μορφή. Προσέξτε: Δεν αναφερόμαστε σε ασφαλή, κρυπτογραφημένα κανάλια επικοινωνίας, τα οποία εκτείνονται από τη συσκευή μας έως εκείνη του ανθρώπου με τον οποίο αλληλογραφούμε. Ενώ αυτά τα κανάλια είναι σίγουρα απαραίτητα και στην πλειονότητα των περιπτώσεων υφίστανται, δεν έχουν καμία σχέση με τη μορφή κατά την οποία αποθηκεύονται τα μηνύματα στους mail servers. Η μορφή αυτή είναι εξ ορισμού μη-κρυπτογραφημένη. Αν έχετε αμφιβολίες επ' αυτού, σκεφτείτε ότι σε διαφορετική περίπτωση δεν θα ήταν δυνατόν να λειτουργούν οι μηχανισμοί anti-SPAM που όλοι οι μεγάλοι email providers έχουν (κι όχι μόνο αυτοί). Αντίθετα, αναφερόμαστε στην κρυπτογράφηση του ίδιου του περιεχομένου των email. Προς το παρόν θα πούμε ότι η η εν λόγω ενέργεια είναι πράγματι εφικτή (και γίνεται δωρεάν). Δεν πρόκειται όμως για την πιο απλή διαδικασία, κι όσοι θέλουν να στέλνουν κρυπτογραφημένα email σίγουρα πρέπει να γνωρίζουν κάποια βασικά πράγματα περί της λεγόμενης κρυπτογραφίας δημοσίου κλειδιού. Όμως όλα όσα αφορούν στην κρυπτογράφηση του email θα έχουμε την ευκαιρία να τα παρουσιάσουμε στο νέο video tutorial που ήδη ετοιμάζουμε για τους συνδρομητές του περιοδικού. Θέλουμε να πιστεύουμε ότι μέσα στο Δεκέμβριο θα είναι διαθέσιμο.

Αυτά προς το παρόν, χαρείτε το χειμώνα και την εορταστική περίοδο που μόλις ξεκίνησε – χαρείτε και το περιοδικό :)

---

1. Παρεμπιπτόντως, το Facebook έχει δικαίωμα να ζητά αληθινά στοιχεία για τους λογαριασμούς των χρηστών, και κάθε φορά που υπάρχει υποψία για fake account δικαιούται να ζητά ταυτοποίηση στοιχείων. Σε περίπτωση μη επιτυχούς ταυτοποίησης, η υπηρεσία μπορεί να προχωρά στο κλείσιμο του αντίστοιχου λογαριασμού. Σκληρός όρος; Μπορεί, από την άλλη όμως ευτυχώς που δεν μας υποχρεώνει κανείς ν' ανοίξουμε λογαριασμό στο Facebook – αληθινό ή fake.



Αυτή την εορταστική σεζόν το delta Hacker κάνει **μεγάλες εκπτώσεις** για όλα τα πακέτα συνδρομών, αλλά δεν σταματά εκεί.
Κληρώνει και πλούσια δώρα όπως **SSD** δίσκους, **iPadAir 2** tablets, **Samsung Galaxy Note 4** smartphones και **MacBook Air** laptops.

Επιλέξτε από το
http://deltahacker.gr/holiday-discounts-and-gifts
τη συνδρομή που σας ταιριάζει
-- και καλή σας επιτυχία!

Hint: Οι εκπτώσεις στα πακέτα συνδρομών φτάνουν το 40%
και κληρώνουμε περισσότερα από ένα SSD drives, tablets, smartphones και laptops.



POST CARD

THE ADDRESS TO BE WRITTEN ON THIS SIDE

Ονέιλ μέιλ

Αγαπητοί μου,

Για άλλη μια φορά θα έπρεπε ν' αναρωτιόμουν από πού να ξεκινήσω την επιστολή μου, αλλά σήμερα δεν είναι όπως τις άλλες φορές. Έχω πολλά να σας πω, και θέλω να αποφύγω το Συνηθισμένο μου Λάθος (TM), να ξεκινάω να σας γράφω για το ένα και το άλλο και να αφήνω το σημαντικό για το τέλος, που πλέον δεν προλαβαίνω.

Έχω λάβει τα μέτρα μου. Πέραν της γραφομηχανής που βρίσκεται μπροστά μου, από τα αριστερά έχω το καινούργιο τάμπλετ της εξαδέλφης με το αρχείο των περιοδικών μας, και από τα δεξιά ένα μικρό δημοσιογραφικό μπλοκάκι κι ένα μολύβι B2.

Έχω ήδη πάρει, ένα προς ένα, τα πιο πρόσφατα τεύχη και έχω φτιάξει μια λίστα με πράγματα στα οποία σας είχα υποσχεθεί ότι θα επανέλθω. Είναι τόσο εκτεταμένη που, τώρα, έτσι όπως την κοιτάω, νοιώθω λίγο άβολα.

Άβολα, αλλά χωρίς ενοχές.

Σημειώνω εδώ, εν παρόδω, πως η ενοχή είναι, κατά την γνώμη μου, ένα ιδιαίτερα προβληματικό και μάλλον υπερεκτιμημένο συναίσθημα. Πολλοί θεωρούν ότι το να νοιώθεις ενοχή είναι δείγμα ικανότητας ενδοσκόπησης και αυτοκριτικής και εκλαμβάνουν, κατ' επέκταση, την έλλειψη ενοχής --την έλλειψη ορατών σημαδιών ενοχικής συμπεριφοράς-- ως δείγμα απάθειας, σκληρότητας και --ναι, καλά το φανταστήκατε-- ουσιαστικής ενοχής (υπό την νομική έννοια του όρου).

Λάθος, αλλά τέλος πάντων, ας το αφήσω στην άκρη. (Το προσθέτω, ωστόσο, στη λίστα μου.)

Σήμερα θα επιστρέψω σε ένα τουλάχιστον από όλα αυτά τα αφημένα στην άκρη θέματα.

Πρόκειται για θέμα που διέτρεχε αρκετές από τις πρόσφατες επιστολές μου: αυτό των τζιτζικιών.

Νόμιζα πως το είχα εξαντλήσει, αλλά με θορύβησε ένα μήνυμα ηλεκτρονικού ταχυδρομείου που μου έστειλε προ ημερών ο ανιψιός, εξ ονόματος του Κυρίου Εκδότου.

Αντί άλλων καθυστερήσεων αντιγράφω εδώ το επίμαχο απόσπασμα.

"*Ναι,* (γράφει ο Κύριος Εκδότης) *καταλαβαίνω και κατανοώ ότι το "τζι τζι" είναι ανθρώπινη περιγραφή. Απλά, δυσκολεύομαι να δεχτώ ότι δεν συμβαίνει *τίποτε* στον αέρα, χωρίς τον παρατηρητή. Κάτι του συμβαίνει του αέρα, ασχέτως αν είμαστε εκεί κοντά ή όχι, ασχέτως αν το ονομάζουμε "τζι τζι" ή "χαριτωμένο μπουρεκάκι", ασχέτως αν μετά συζητάμε για το αν πράγματι του συμβαίνει κάτι (του αέρα, εξαιτίας του τζιτζικιού), όταν εμείς είμαστε ή δεν είμαστε εκεί για να ακούσουμε το "τζι τζι" και τόσο αναπάντεχα (ή όχι) να προβληματιστούμε κ.λπ.*"

"*Να πω και κάτι άλλο* (συνεχίζει). *Θα εκρηγνυόταν το* ***Bárðarbunga****, αν δεν υπήρχε κανείς --Ισλανδός ή whatever-- να το παρατηρήσει; θα κινδύνευαν να λιώσουν οι παγετώνες και να προκληθούν πλημμύρες στην περιοχή (...) αν κανείς δεν είχε παρατηρήσει ότι το* ***Bárðarbunga*** *δραστηριο-*

ποιήθηκε, κάνοντας αυτό που ξέρει να κάνει ανεξαρτήτως του παρατηρητή;"

Οφείλω να παραδεχτώ πως τα ερωτήματα του Κυρίου Εκδότου με ταρακούνησαν. Ειδικά τα σχόλια περί "χαριτωμένου μπουρεκιού". Ένοιωσα πως απέτυχα να μεταδώσω τον πυρήνα του επιχειρήματός μου, και νοιώθω την ανάγκη να επανέλθω.

Λοιπόν, υπάρχουν δύο τρόποι να προχωρήσει κανείς από εδώ. Ο σκληροπυρηνικός και ο διαλλακτικός. Στην πραγματικότητα δεν διαφέρουν και πολύ.

Ο σκληροπυρηνικός τρόπος θα ήταν να εμείνω στο επιχείρημά μου, αδιαφορώντας για τα άτοπα στα οποία δείχνει, φαινομενικώς, να οδηγεί. Θα έλεγα κάτι σαν το παρακάτω, ας πούμε.

«"Ισλανδία"; Τι θα πει "Ισλανδία"; Ούτε Ισλανδία, ούτε **Bárðarbunga** υπάρχουν, αν δεν υπάρχουν ήδη οι Ισλανδοί, όπως ακριβώς και το "τζι τζι" δεν ακούγεται αν δεν είμαστε εμείς εκεί να το ακούσουμε.»

Από την αυστηρώς λογική πλευρά του επιχειρήματός μου, θα έχω απαντήσει και μάλιστα θα έχω δίκιο: Εφόσον κατά τον ορισμό μου το ρήμα "υπάρχω" αποκτά το χαρακτήρα του υπαρξιακού ποσοδείκτη μόνο κατά το μέτρο που υπάρχουν άνθρωποι να το χρησιμοποιήσουν, εξυπακούεται πως αν δεν υπάρχουν Ισλανδοί, δεν υπάρχει ούτε το **Bárðarbunga.** Και αντιστοίχως, αν το "ακούγομαι" το εκλαμβάνω ως έχον νόημα μόνο κατά το μέτρο που υπάρχουν άνθρωποι οι οποίοι ακούνε, εξυπακούεται πως αν δεν υπάρχουν αυτοί, τότε ούτε τζι τζι ακούγεται.

Όπερ έδει δείξαι.

Υποθέτω, όμως, ότι αυτή η απάντηση, λογικώς ορθή έστω, δεν φαντάζει ικανοποιητική. ΟΚ, λοιπόν. Θα το προσεγγίσω αλλιώς.

Η βασικότερη αντίρρηση που διατυπώνεται είναι πως δεν μπορώ να ισχυρίζομαι στα σοβαρά ότι κοτζάμ ηφαίστειο εμφανίζεται και εξαφανίζεται ανάλογα με το αν υπάρχουν άνθρωποι εκεί γύρω να το δουν και να μιλήσουν γι' αυτό, ή ότι μυριάδες τζιτζίκια περιμένουν σινιάλο πότε θα σκάσουμε μύτη, και τότε και μόνο τότε αρχινάνε όλα μαζί το "τζι τζι".

Εύλογη η αντίρρηση, μόνο που εγώ δεν ισχυρίστηκα τίποτα για το τι κάνει το ηφαίστειο ή το τζιτζίκι.

Για να το δούμε λίγο: Όταν λέμε "κάτι" για "κάτι", αυτό το "κάτι" δε φτάνει ποτέ σε μας αδιαμεσολάβητο. Πάντα σχετίζεται με τον τρόπο με τον



οποίο βλέπουμε και κατανοούμε τα πράγματα. Και τα πράγματα τα βλέπουμε και τα κατανοούμε σύμφωνα με ένα ήδη δεδομένο πλαίσιο ερμηνείας που δεν προκύπτει από τα ίδια τα πράγματα: είναι πάντα ανθρώπινη κατασκευή. Αυτό είναι ο πυρήνας του επιχειρήματος και σας παραπέμπω, αν έχετε όρεξη, σε αυτά που λέγαμε το καλοκαίρι για την θεωρία περί αλήθειας του Τάρσκι (αλλά και των σχολαστικών φιλοσόφων του Μεσαίωνα) (επιστολή #034), καθώς και στη συζήτηση που είχα με τη θεία και τον ανιψιό γι' αυτό ακριβώς το θέμα, το ρήμα "είναι" (επιστολή #035). Αν θυμάστε, τότε ήταν που πρωτοτέθηκε το ζήτημα με τα τζιτζίκια.

Μια λύση, για να αποφεύγω κακοτοπιές, θα ήταν να μην χρησιμοποιώ πλέον το ρήμα "υπάρχω" ή το ρήμα "ακούγομαι". Θα μπορούσα, για να αποδώσω τη λεπτή απόχρωση που έχω κατά νου, να χρησιμοποιώ εφεξής το ρήμα "*υπάρχω" ή το ρήμα "*ακούγομαι" (με αστερίσκο). Παραδείγματα:

"Αν δεν *υπάρχουν Ισλανδοί, δεν *υπάρχει ούτε το **Bárðarbunga**."

"Αν δεν *υπάρχουν άνθρωποι, το 'τζι τζι' δεν *ακούγεται." Και ούτω καθεξής.

Καλύτερα τώρα; Εστω. Αλλά πείτε μου, σε τι έγκειται η διαφορά ενός ρήματος με αστερίσκο από το ίδιο ρήμα χωρίς αστερίσκο;

Στο ρήμα με αστερίσκο, ο αστερίσκος παίζει ρόλο υπενθύμισης και υπογράμμισης της ανθρώπινης παρουσίας. Ενώ στο ρήμα χωρίς αστερίσκο, η ανθρώπινη παρουσία πηγαίνει στο παρασκήνιο. Ωστόσο δεν εξαφανίζεται. Είναι πάντα εκεί. Υπορρήτως, αλλά σαφώς.

Αυτό είναι το λεπτό σημείο που συνήθως παραβλέπουμε. Ο αστερίσκος είναι για υπενθύμιση. Δεν προσθέτει ούτε αφαιρεί κάτι. Δεν αλλάζει τίποτα στο ρήμα. Ακόμα και όταν λέμε ότι "κάτι είναι", αυτό που εννοούμε είναι πως "κάτι *είναι".

ΟΚ, θα μου πείτε. Πού μας οδηγούν όλα αυτά; Γιατί ταλαιπωρούμαστε;

Να σας πω γιατί ταλαιπωρούμαστε. Ταλαιπωρούμαστε, αγαπητοί μου, γιατί ως άνθρωποι είμαστε, συνήθως, επιρρεπείς σε δύο σημαντικά λάθη. Το πρώτο είναι πως μπερδεύουμε ή παραβλέπουμε τη διαφορά μεταξύ περιγραφικών και οντολογικών αποφάνσεων. Κάνουμε το ένα και νομίζουμε πως κάνουμε το άλλο. Λέμε, ας πούμε, για άνοδο της θερμοκρασίας και νομίζουμε ότι στην φύση υπάρχει κάτι που (θα μπορούσε να) λέγεται θερμοκρασία, η τιμή του οποίου αυξάνεται. Στην φύση βέβαια δεν υπάρχουν θερμοκρασίες, υπάρχει μόνο η στατιστική εκτίμηση της μέσης κινητικής ενέργειας κάποιων μορίων.

Και το δεύτερο είναι ότι θεωρούμε πως οι οντολογικές αποφάνσεις είναι δυνατές ενώ δεν είναι. Period. Δεν μπορούμε ποτέ να μιλήσουμε για το "τι είναι" και "τι δεν είναι". Ακόμα και όταν μιλάμε για κινητική ενέργεια μορίων, δεν αναφερόμαστε σε κάποια "αντικειμενική ιδιότητα" των μορίων. Ούτε η θερμοκρασία, αλλά ούτε η κινητική ενέργεια των μορίων ως τέτοια υπάρχουν "αντικειμενικά". Υπάρχουν μόνο ως έννοιες, και μόνο στο πλαίσιο ενός μοντέλου ερμηνείας του κόσμου. Είναι αυτό, το μοντέλο δηλαδή, που τους δίνει νόημα και περιγράφει τις αλληλεξαρτήσεις τους με τις άλλες έννοιες του ίδιου μοντέλου.

Εκτός μοντέλου, όλες μας οι έννοιες είναι α-νόητες.

Κάτι σαν το "τζι τζι" δηλαδή.

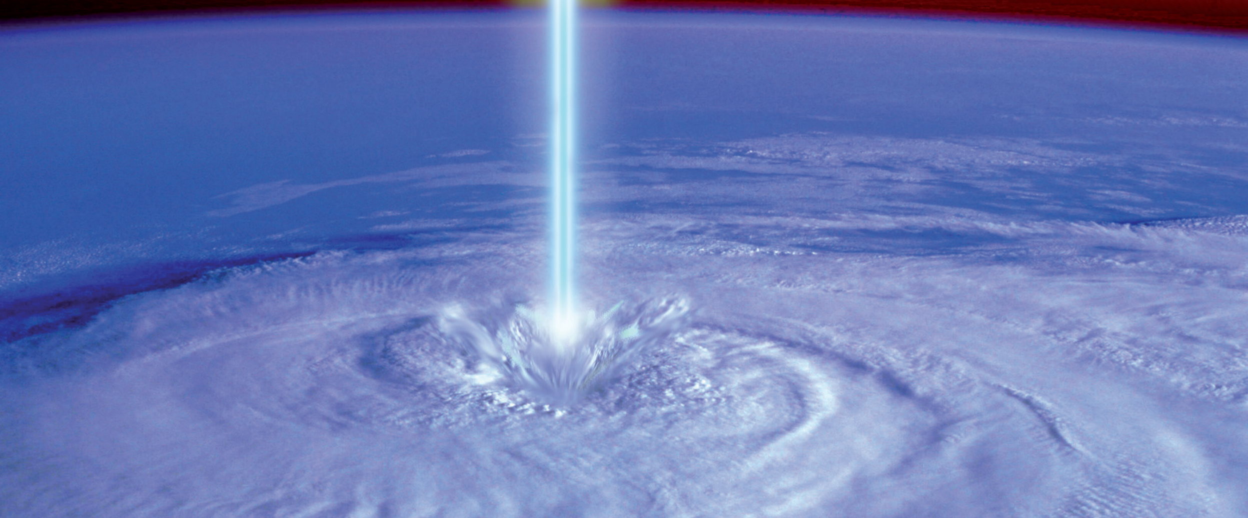
Αλλά φοβάμαι ότι μου τελειώνει ο χώρος, οπότε ας μείνουμε εδώ. Το συμπληρώνω ήδη στην λίστα μου, όμως, και θα συνεχίσω (ελπίζω) την επόμενη φορά.

Σας ασπάζομαι,
Θείος Ακάκιος

ονέμλ μέλ

Skill: Intermediate
Tags: DDoS, LOIC, HOIC, iptables

Low Orbit Ion Cannon



by Spir@l Evolut10n

Ανελέητες επιθέσεις για καλύτερη άμυνα! [Μέρος 1/2]

Πριν από περίπου δυο χιλιάδες χρόνια, ο Κινέζος στρατηγός Σουν Τζου έγραφε ότι η επίθεση αποτελεί το κλειδί στην άμυνα και η άμυνα αποτελεί το σχεδιασμό της επίθεσης. Εμείς, ως system administrators, δεν σκοπεύουμε να κηρύξουμε τον πόλεμο σε κανέναν. Θέλουμε όμως να γίνουμε τιτάνες στην άμυνα. Για λόγους εξάσκησης, λοιπόν, θα επιτεθούμε αλύπητα στον εαυτό μας.

Κάθε φορά που καταπιανόμαστε μ' ένα είδος επιθέσεων φροντίζουμε να παρουσιάσουμε και τα αντίστοιχα μέτρα προστασίας. Εξάλλου, δεν είναι λίγες οι φορές που έχουμε ασχοληθεί αποκλειστικά με τον έναν ή τον άλλο μηχανισμό ασφαλείας. Για παράδειγμα, στο τεύχος 036 αναφερθήκαμε στο περίφημο σύστημα ανίχνευσης εισβολών (Intrusion Detection System) ονόματι Snort, ενώ δυο μήνες νωρίτερα (τεύχος 034) είχαμε παρουσιάσει το fail2ban. Γενικά οι αναφορές μας στην ασφάλεια δεν έχουν τελειωμό, αφού πολύ απλά δεν θέλουμε να απουσιάζουν από το περιεχόμενο του περιοδικού.

Είναι όμως αυτό αρκετό; Όχι. Μια παρουσίαση, όσο ισορροπημένη κι αν είναι μεταξύ θεωρίας και πράξης και σ' όσο βάθος κι αν φτάνει, δεν θα καταργήσει ποτέ την αξία της πρακτικής ενασχόλησης μ' ένα αντικείμενο. Αυτό το τονίζουμε σε κάθε ευκαιρία, αλλά η ίδια η επισήμανση δεν παύει να 'ναι άλλη μία θεωρητική συμβουλή ;) Κάπως έτσι, φίλες και φίλοι, αποφασίσαμε να καταπιαστούμε ξανά με ένα από τα αγαπημένα μας χόμπι, το οποίο βοηθά στην καλύτερη κατανόηση των συστημάτων και στην απόκτηση ουσιαστικών γνώσεων. Αυτό το χόμπι δεν είναι άλλο από την ίδια την πρακτική εξάσκηση: τόσο όσον αφορά στην επίθεση, όσο και στην άμυνα. Πού ξέρετε; Ίσως έτσι ενθαρρύνουμε ακόμα και τους πιο θεωρητικούς τύπους να σηκώσουν τα μανίκια και ν' ακολουθήσουν τα βήματά μας. Αν το κάνουν, ξέρουμε ότι δεν θα το μετανιώσουν.



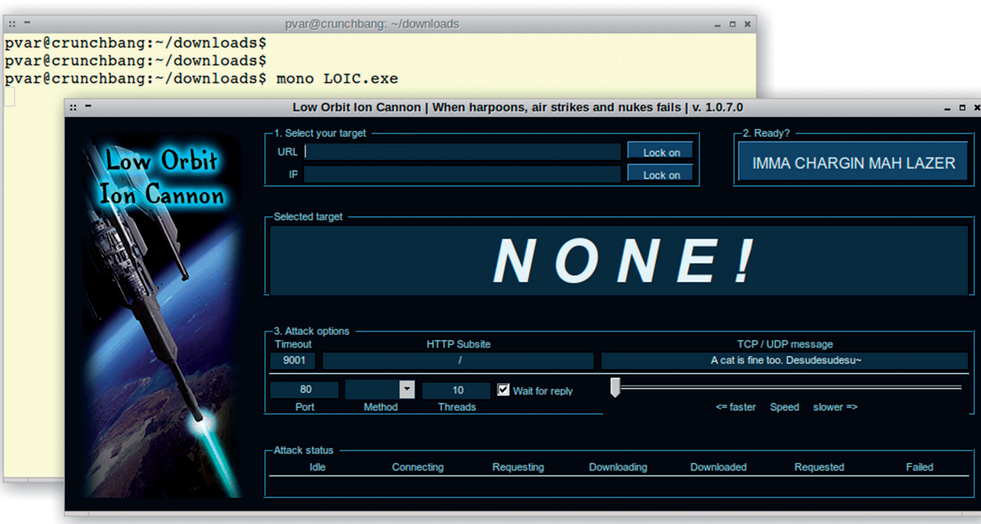
Το περιβάλλον του LOIC είναι εξαιρετικά απλό, αλλά όχι και λιτό. Η αισθητική του παραπέμπει σε αυτό που ονομάζουμε "καγκουριλίκι". Στην εικόνα ετοιμαζόμαστε να εξαπολύσουμε μια επίθεση UDP flood στην εικονική μηχανή με τον web server.

Εικονικό εργαστήριο

Σε αυτό το άρθρο θα μιλήσουμε για ένα μικρό τοπικό δίκτυο, για τις επιθέσεις που πραγματοποιήσαμε και τους καρπούς της εργασίας μας. Το δίκτυο διέθετε μόλις τρία μέλη, δύο από τα οποία ήταν εικονικές μηχανές. Η μία έπαιζε το ρόλο του επιτιθέμενου και η άλλη το ρόλο του αμυνόμενου. Τώρα θα πούμε κάτι που ενδέχεται να σας προκαλέσει σκοτοδίνη, αλλά πρέπει να μας εμπιστευθείτε: Η εικονική μηχανή για τις επιθέσεις έτρεχε Windows :D Η επιλογή αυτή ήταν επιβεβλημένη από το στόχο που

είχαμε θέσει αρχικά, και στον οποίο θα αναφερθούμε σύντομα. Το δεύτερο εικονικό σύστημα έτρεχε το Crunchbang Linux και ήταν στημένο ως web server. Εκεί φιλοξενούσαμε ένα αντίγραφο του Parabing, το οποίο επισκεπτόμασταν από το τρίτο μηχάνημα του δικτύου, για να ελέγχουμε την κατάσταση του web server. Παρεμπιπτόντως, το τρίτο μέλος του δικτύου ήταν ο ίδιος ο host των εικονικών μηχανών. Τέλος, σημειώστε ότι για τις εικονικές κάρτες δικτύου είχαμε επιλέξει το "bridged networking".

Ας δούμε τώρα το στόχο των πειραμάτων μας, για να εξηγήσουμε και την απροσδόκητη προτίμηση στα Windows. Θα περίμενε κανείς ότι στο ρόλο του επιτιθέμενου θα βάζαμε το Kali Linux και θα αξιοποιούσαμε κάποια δημοφιλή πλατφόρμα επιθέσεων. Περιττό να πούμε ότι διαθέταμε προ πολλού ένα τέτοιο εικονικό σύστημα, ωστόσο επιλέξαμε να μην το χρησιμοποιήσουμε καθόλου. Στα πειράματά μας, βλέπετε, θέλαμε να επιτεθούμε με τη βοήθεια του διαβόητου Low Orbit Ion Cannon (LOIC). Έτσι, θα είχαμε την ευκαιρία να αναμετρηθούμε με ένα δικτυακό "όπλο" που μπορεί να μην είναι το πιο επικίνδυνο, αλλά είναι σίγουρα ξακουστό και στο παρελθόν είχε προκαλέσει μεγάλους μπελάδες. Μέσα από αυτή τη διαδικασία θέλαμε να γνωρίσουμε καλύτερα το iptables του Linux και να εξετάσουμε πιθανές λύσεις για την αντιμετώπιση των επιθέσεων Denial of Service (DoS). Το LOIC κατασκευάστηκε περί το 2008 από μέλη των Anonymous. Το εργαλείο πραγματοποιεί επιθέσεις DoS (Denial of Service) και δημιουργήθηκε για να διανεμηθεί στους ανά τον κόσμο εθελοντές "ακτιβιστές" (Σ.τ.Ε. Από τις ακριβέστερες χρήσεις των εισαγωγικών που έχουμε δει.) Με αυτόν τον τρόπο θα συγκροτούταν στα γρήγορα ένας μικρός (;) στρατός, που θα μπορούσε να εξαπολύει επιθέσεις DDoS (Distributed Denial of Service). Βέβαια, οι εθελοντές που στήριζαν τις πρωτοβουλίες των Anonymous είχαν συνήθως χαμηλό επίπεδο τεχνικών γνώσεων κι ως εκ τούτου το LOIC έπρεπε να 'ναι ιδιαίτερα εύχρηστο. Για τον ίδιο λόγο γράφτηκε σε .NET και τρέχει στα Windows.



Το LOIC είναι γραμμένο σε .NET κι ως εκ τούτου μπορεί να τρέξει και στο Linux, με τη βοήθεια της βιβλιοθήκης MONO. Πάντως, η συμπεριφορά του στο Linux ήταν ασταθής και για τις δοκιμές μας επιλέξαμε τη σταθερότητα των Windows (ξέρουμε κι εμείς να γινόμαστε trolls ;))

Γνωριμία με τα όπλα

Όταν φτιάξαμε το τοπικό δίκτυο και σηκώσαμε τον web server στο εικονικό μηχάνημα-στόχο, ξεκινήσαμε αμέσως τις επιθέσεις. Θέλαμε να διαπιστώσουμε αν το Low Orbit Ion Cannon ήταν τόσο αποτελεσματικό όσο λεγόταν. Φυσικά, μια επίθεση DoS είναι αδύνατο να αποδειχθεί αποτελεσματική όταν πραγματοποιείται μέσω Internet, από μια τυπική οικιακή γραμμή. Αυτό συμβαίνει γιατί το upstream bandwidth σε μια τέτοια σύνδεση είναι μερικές τάξεις μεγέθους μικρότερο από το downstream bandwidth ενός οποιουδήποτε server Εκεί Έξω (TM). Επομένως, ο δράστης δεν θα μπορέσει να "στραγγαλίσει" ποτέ τη σύνδεση του θύματος. Εμείς όμως δεν ανησυχούσαμε καθόλου γι' αυτό, αφού η επίθεση θα διεξαγόταν σε επίπεδο τοπικού δικτύου. Αυτό που μας έκαιγε ήταν αν το ένα στιγμιότυπο του LOIC θα αποδεικνυόταν ικανό να "στουμπώσει" τον web server. Όπως καταλαβαίνετε, δεν είχαμε καμία όρεξη να στήσουμε μερικές δεκάδες εικονικές μηχανές με Windows, μόνο και μόνο για να έχουμε ένα ισάριθμο πλήθος στιγμιότυπων του LOIC ;) Για καλή μας τύχη, στις συνθήκες που είχαμε δημιουργήσει, το προγραμματάκι των Anonymous αποδείχθηκε άκρως αποτελεσματικό!

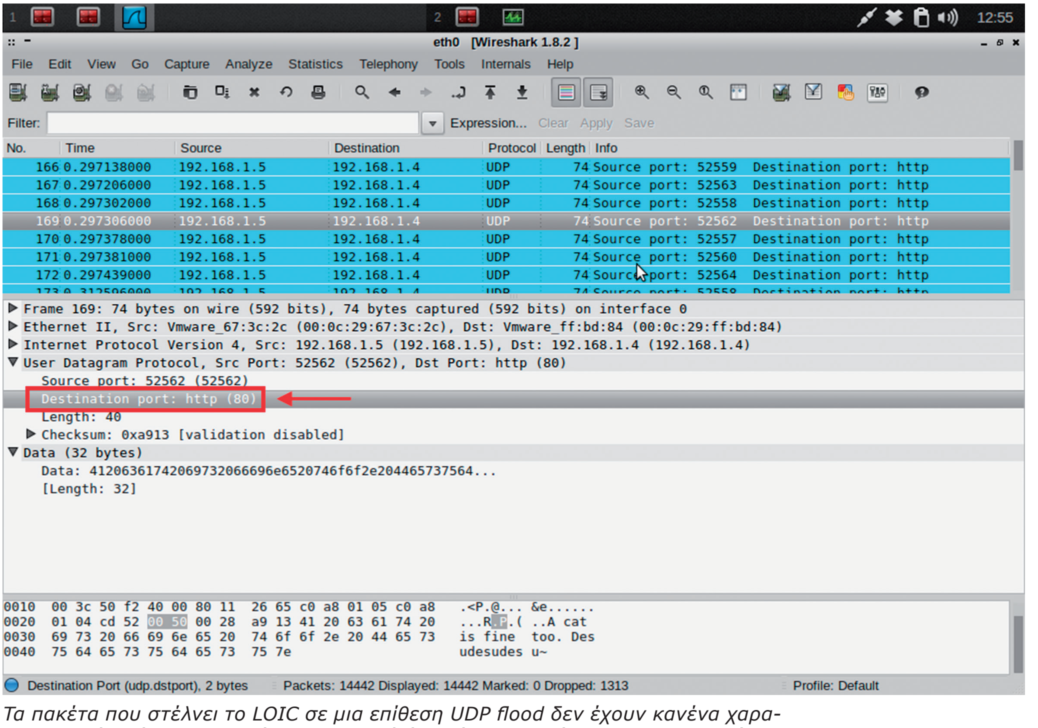
Το περιβάλλον του LOIC είναι εξαιρετικά απλό και οι ελάχιστες ρυθμίσεις του δεν χρειάζονται καμία επεξήγηση. Αυτό που αξίζει να σημειώσουμε είναι ότι το πρόγραμμα μπορεί να πραγματοποιεί UDP, TCP και HTTP flood attacks. Με απλά λόγια, μπορεί να δημιουργεί και να στέλνει εκατοντάδες (αν όχι χιλιάδες) αιτήματα σε κάποιο από τα τρία πρωτόκολλα, σε ελάχιστο χρόνο και αδιάκοπα. Με αυτόν τον τρόπο επιχειρεί να εξαντλήσει τους πόρους του απομακρυσμένου συστήματος ή της εκάστοτε υπηρεσίας, ώστε να μην υπάρχει κανένα περιθώριο για την εξυπηρέτηση των αθώων (κανονικών) επισκεπτών/χρηστών. Εμείς δοκιμάσαμε και τις τρεις παραλλαγές της επίθεσης και σε κάθε περίπτωση ο web server έπεφτε με ευκολία.

Εφόσον οι επιθέσεις μας ήταν πετυχημένες, θα μπορούσαμε να προχωρήσουμε αμέσως στον πρωταρχικό μας στόχο: Στην αναζήτηση τεχνικών για την αντιμετώπιση των επιθέσεων DoS, με το iptables. Ωστόσο, έχοντας τη διάθεση να δυσκολέψουμε το παιχνίδι, ξεκινήσαμε μια αναζήτηση για μεθόδους που θα αύξαναν την σφοδρότητα των επιθέσεων. Έτσι πέσαμε πάνω στη μετεξέλιξη του LOIC, που είναι το επίσης διάσημο HOIC (High Orbit Ion Canon). Αυτό το πρόγραμμα διεξάγει αποκλειστικά HTTP flood attacks, αλλά περιλαμβάνει μια δυνατότητα που το καθιστά ισχυρότερο έναντι του LOIC. Αναφερόμαστε στα λεγόμενα booster scripts. Τα εν λόγω scripts είναι γραμμένα σε VB script[1] και περιλαμβάνουν λίστες με διάφορες σελίδες που εξυπηρετεί το μηχάνημα - στόχος, τυχαίους referrers και τυχαία user agent strings. Αξιοποιώντας ένα τέτοιο script, το HOIC μπορεί να δημιουργεί αιτήματα HTTP που διαφοροποιούνται σε μεγάλο βαθμό. Έτσι, οι συνδέσεις μοιάζουν να προέρχονται από διαφορετικούς χρήστες, γεγονός που προκαλεί σύγχυση σε διάφορα συστήματα προστασίας. Πάντως, αν και δεν αντισταθήκαμε στον πειρασμό να δοκιμάσουμε το HOIC, η αλήθεια είναι δεν ασχοληθήκαμε ιδιαίτερα μαζί του. Εμείς βλέπετε θα εξετάζαμε λύσεις που στηρίζονται στο iptables. Ως γνωστόν, αυτές οι λύσεις δεν εστιάζουν στα δεδομένα που μεταφέρουν τα πακέτα, ούτε και στο περιεχόμενο των header των HTTP requests[2].

Τέλος, να πούμε ότι τα δύο εργαλεία των Anonymous τα κατεβάσαμε από τις σε-

1. Ουφ! Αυτοί οι Anonymous πέφτουν συνεχώς στα μάτια μας ;)
2. Δεν είναι αδύνατο κάτι τέτοιο. Το iptables έχει τη δυνατότητα να πραγματοποιεί αναζητήσεις μέσα στο περιεχόμενο των πακέτων, για τον εντοπισμό ενός οποιουδήποτε string. Ωστόσο, το φιλτράρισμα των πακέτων με βάση την έκβαση μιας αναζήτησης εισάγει μεγάλες καθυστερήσεις κι επομένως αποφεύγεται.

λίδες http://sourceforge.net/projects/loic και http://sourceforge.net/projects/highorbitioncannon. Πρέπει όμως να επιστήσουμε και την προσοχή σας! Τα δύο προγράμματα φημολογείται ότι περιλαμβάνουν backdoor, που επιτρέπουν τη μετατροπή του μηχανήματός μας σε zombie (μέλος κάποιου botnet). Η φήμη αυτή αποτελεί ισχυρό κίνητρο για να μη χρησιμοποιήσετε αληθινό (φυσικό) μηχάνημα αλλά να στραφείτε σε κάποια εικονική μηχανή ;)

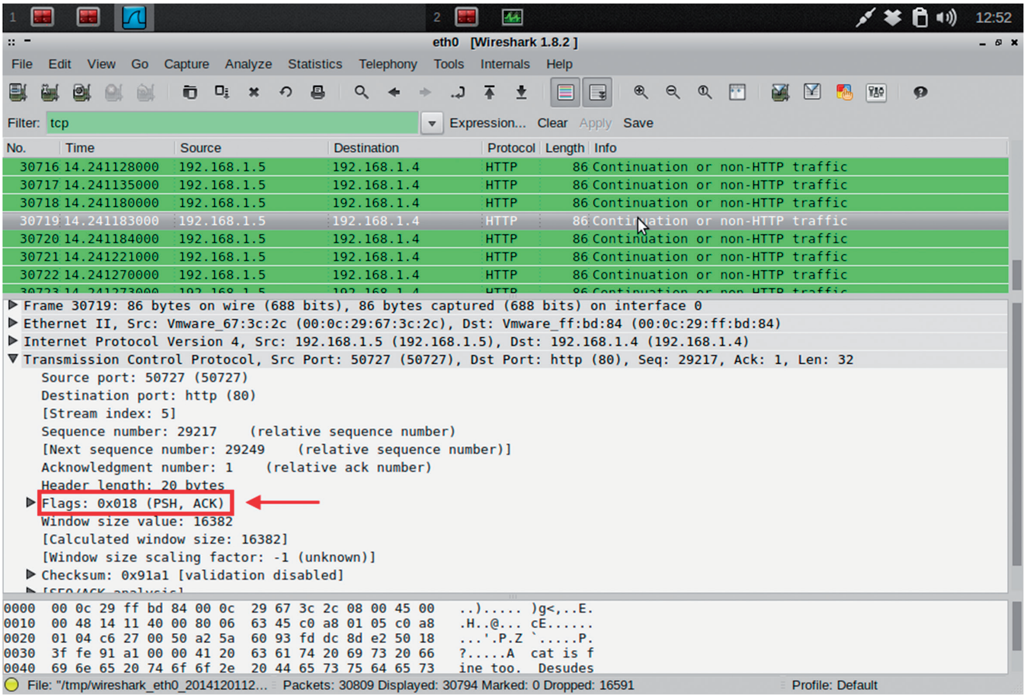


Τα πακέτα που στέλνει το LOIC σε μια επίθεση UDP flood δεν έχουν κανένα χαρακτηριστικό γνώρισμα. Το μόνο τους κοινό ήταν ότι κατευθύνονταν προς τη θύρα 80, αλλά αυτό θα μπορούσε να αλλάξει πανεύκολα από το περιβάλλον του LOIC.

Μάθημα ανατομίας

Πριν καταπιαστούμε με την άμυνα, έπρεπε να γνωρίσουμε τον εχθρό. Μπορεί να είχαμε υπόψη μας ότι το LOIC έστελνε ένα τεράστιο σύνολο αιτημάτων, αλλά δεν είχαμε ιδέα για το πώς μπορούμε να διακρίνουμε τα σχετικά πακέτα. Γι' αυτό το λόγο επιστρατεύσαμε το Wireshark. Αφού ξεκινήσαμε την καταγραφή της κίνησης στο τοπικό δίκτυο, εξαπολύσαμε αρχικά μια επίθεση UDP flood. Το LOIC έστελνε πακέτα UDP στη θύρα 80, αλλά δεν είχαν κανένα ιδιαίτερο γνώρισμα. Επιπρόσθετα, γνωρίζαμε ότι ακόμα και η δικτυακή θύρα μπορούσε να αλλάξει πανεύκολα, από το περιβάλλον του LOIC. Υπό άλλες συνθήκες, το γεγονός αυτό θα μας προκαλούσε ιδιαίτερο προβληματισμό. Πώς μπορείς να φιλτράρεις τα κακόβουλα πακέτα, όταν δεν έχουν κανένα ιδιαίτερο γνώρισμα; Σ' αυτή την περίπτωση, όμως, το ζήτημα ήταν από τη φύση του απλό. Η επίθεση που εξετάζαμε αφορούσε σε ένα πρωτόκολλο, μέσω του οποίου δεν παρέχεται καμία υπηρεσία από το σύστημά μας. Συμπέρασμα; Σε περίπτωση κατάχρησης θα μπορούσαμε να μπλοκάρουμε προσωρινά όλα τα εισερχόμενα πακέτα UDP.

Στη συνέχεια, με το Wireshark να καταγράφει τα πάντα, πραγματοποιήσαμε μια επίθεση TCP flood. Εδώ τα ευρήματά μας παρουσίαζαν περισσότερο ενδιαφέρον. Πέρα από το μεγάλο ρυθμό με τον οποίο κατέφθαναν τα πακέτα, παρατηρήσαμε ότι τα περισσότερα είχαν ενεργοποιημένα τα PSH και ACK flags. Το πρώτο από αυτά ζητά το άμεσο άδειασμα της cache του TCP layer, του server. Ας εξηγήσουμε τι σημαίνει αυτό σε απλά ελληνικά. Η υλοποίηση του TCP layer ενσωματώνει έναν προσωρινό αποθηκευτικό χώρο για κάθε σύνδεση. Αυτός ο χώρος εξασφαλίζει μια ομαλότητα στην επικοινωνία των "στρωμάτων" του λογισμικού, ενώ επιτρέπει στα διακινούμενα πακέτα να μεταφέρουν το μέγιστο δυνατό όγκο δεδομένων και να μην ταξιδεύουν μισοάδεια. Αυτή η λειτουργία είναι αρκετά χρήσιμη, αλλά σε ορισμένες περιπτώσεις γίνεται ανεπιθύμητη! Σε μια σύνδεση SSH, για παράδειγμα, τα μηνύματα που παράγει ο server θέλουμε να μεταδίδονται αμέσως και όχι όταν θα γεμίσει κάποιος μυστηριώδης αποθηκευτικός χώρος, αγνώστου χωρητικότητας. Για αυτές τις περιπτώσεις προσφέρεται το flag PUSH (ή PSH), που επιβάλει το άμεσο άδειασμα της cache του TCP layer. Το δε ACK flag (από το ACKNOWLEDGED) χρησιμοποιείται σε πακέτα του TCP, όταν ο παραλήπτης προηγούμενων πακέτων θέλει να ενημερώσει τον αποστολέα για τη σωστή παραλαβή τους. Όπως καταλαβαίνετε, η χρήση αυτών των δύο flags στα πακέτα του LOIC είναι παραπλανητική. Εκτός του ότι τα πακέτα καταφτάνουν με καταιγιστικό ρυθμό, τα σχετικά flags επιβεβαιώνουν τη λήψη πακέτων που δεν έστειλε ποτέ ο server και ζητούν την αποστολή νέων. Εν ολίγοις, αυτός ο τρόπος χρήσης των συγκεκριμένων flags προκαλεί σύγχυση.



Να και κάτι που μοιάζει περισσότερο χρήσιμο! Όταν το LOIC πραγματοποιεί επιθέσεις TCP flood, στέλνει πακέτα με τα PSH και ACK flags. Ίσως αυτό διευκολύνει τον εντοπισμό των επιθέσεων.

Για το τέλος αφήσαμε την εξέταση της επίθεσης HTTP flood. Το πρωτόκολλο HTTP υλοποιείται "πάνω" από το TCP. Με άλλα λόγια, οι συνδέσεις HTTP πραγματοποιούνται πάντοτε μέσα στο πλαίσιο μιας σύνδεσης TCP. Εξετάζοντας τα διακινούμενα πακέτα με το Wireshark, διαπιστώσαμε ότι δεν διέθεταν κανένα ιδιαίτερο γνώρισμα. Αν εξαιρέσουμε το μεγάλο ρυθμό με τον οποίο πραγματοποιούνταν νέες συνδέσεις, στο επίπεδο του TCP δεν εντοπίζαμε κανένα ιδιαίτερο γνώρισμα – τίποτα που να διακρίνει τα πακέτα του LOIC από τα υπόλοιπα. Ο προβληματισμός μας άρχισε να φουντώνει, μέχρι τη στιγμή που παρατηρήσαμε το περιεχόμενο των αιτημάτων HTTP. Για κάποιο λόγο, τα αιτήματα HTTP που σχηματίζει το LOIC έχουν ασυνήθιστη μορφή και διακρίνονται αμέσως. Για την ακρίβεια, το header τους ξεκινά με το ακόλουθο string: GET /HTTP/1.0\r\n. Η παρουσία των δύο χαρακτήρων στο τέλος (CARRIAGE RETURN και NEW LINE) δεν προβλέπεται κι επί της ουσίας συνιστά συντακτικό σφάλμα[3]. Αυτό ήταν, λοιπόν! Αναζητώντας το συγκεκριμένο string στα HTTP requests, θα εντοπίζαμε εύκολα τις συνδέσεις του LOIC και θα τις κόβαμε μια και καλή. Όπως καταλαβαίνετε, σε αυτό το σημείο είχαμε αποφασίσει να κλέψουμε λίγο! Θα πείραζε πολύ αν επιστρατεύαμε το fail2ban για το μπλοκάρισμα αυτών των επιθέσεων; Στο κάτω-κάτω το συγκεκριμένο πρόγραμμα στηρίζεται στις πλάτες του iptables...

Πρώτη προσέγγιση

Έχοντας συγκεντρώσει όλες τις παρατηρήσεις μας, κλείσαμε το Wireshark και ανοίξαμε ένα τερματικό. Πιστεύαμε ότι έχουμε βρει τη λύση για όλα τα προβλήματα και ετοιμαζόμασταν να γράψουμε ένα script για την κατάλληλη ρύθμιση του iptables. Τα πράγματα όμως δεν ήταν τόσο απλά. Είχαμε κάνει ένα ύπουλο λογικό σφάλμα και θα το διαπιστώναμε πολύ σύντομα. Περιττό να πούμε ότι η πορεία που ακολουθήσαμε για την οριστική λύση δεν ήταν ακριβώς ευθύγραμμη, αλλά πιστεύουμε ότι αξίζει να την παρακολουθήσετε. Ας δούμε για την ώρα πώς ξεκινήσαμε. Ακολουθεί το script για τη ρύθμιση του iptables:

```
iptables -F
iptables -X
iptables -N inspect
iptables -N log_n_drop
iptables -P INPUT DROP
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth0 -j inspect
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -P OUTPUT ACCEPT
iptables -A OUTPUT -m state --state INVALID -j DROP
iptables -A OUTPUT -j inspect
iptables -A log_n_drop -j LOG --log-prefix '[suspected DoS] '
```

3. Παρεμπιπτόντως, οι συγκεκριμένοι δύο χαρακτήρες χρησιμοποιούνται στα Windows για το μαρκάρισμα του τέλους μιας γραμμής. Πφ! Πολύ πρόχειρη δουλειά αυτό το LOIC, τελικά.

```
13 iptables -A log_n_drop -j DROP
14 # block UDP flood
15 iptables -A inspect -p udp -m udp --dport 80 \
         -m limit --limit 1/sec --limit-burst 8 -j ACCEPT
16 iptables -A inspect -p udp -j log_n_drop
17 # block TCP PSH,ACK flood
18 iptables -A inspect -p tcp -m tcp --tcp-flags ALL PSH,ACK \
         -m limit --limit 1/sec --limit-burst 8 -j ACCEPT
19 iptables -A inspect -p tcp -m tcp --tcp-flags ALL PSH,ACK \
         -j log_n_drop
```

Το script ξεκινά με τον πλέον κλασικό τρόπο για το είδος του: Με τη διαγραφή όλων των κανόνων και των προσαρμοσμένων (custom) αλυσίδων του iptables. Στη συνέχεια δημιουργούνται δύο νέες αλυσίδες. Η μία ονομάζεται inspect και θα περιλαμβάνει τους κανόνες για τον εντοπισμό και το φιλτράρισμα των επιθέσεων DoS. Η δεύτερη αλυσίδα ονομάζεται log_n_drop και πιστεύουμε ότι η αποστολή της είναι προφανής ;) Τα πακέτα που θα καταλήγουν σε αυτήν θα καταγράφονται στο log του iptables κι ακολούθως θα απορρίπτονται.

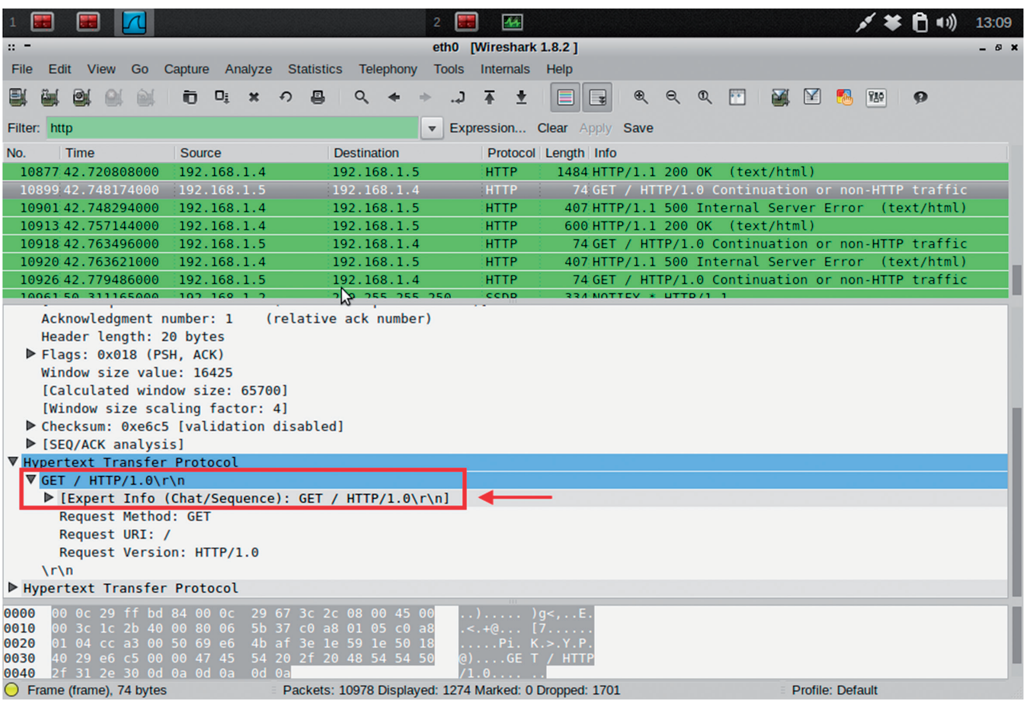
Στις γραμμές 05 έως 08 πραγματοποιείται η ρύθμιση της αλυσίδας INPUT. Αρχικά ορίζουμε τη γενική πολιτική (policy) της αλυσίδας, που μπορείτε να την αντιλαμβάνεστε σαν την προεπιλεγμένη ενέργεια. Όπως έχουμε ξαναπεί, μια καλή τακτική για τη ρύθμιση ενός firewall είναι το κλείσιμο όλων των "περασμάτων" και στη συνέχεια το άνοιγμα εκείνων μόνο που χρειαζόμαστε. Με αυτόν τον τρόπο εξασφαλίζουμε ότι το firewall θα έχει μόνο τα απολύτως απαραίτητα "ανοίγματα" και κανένα άλλο. Γι' αυτό το λόγο επιλέξαμε σαν γενική πολιτική το DROP. Ακολούθως εξασφαλίζουμε ότι τα πακέτα που προέρχονται από το ίδιο το σύστημα (interface lo) θα γίνονται εξ ορισμού αποδεκτά, ενώ εκείνα που προέρχονται από τον έξω κόσμο (interface eth0) θα προωθούνται για έλεγχο στην αλυσίδα inspect. Όσα πακέτα διέρχονται από την αλυσίδα inspect χωρίς να απορριφθούν ή να προωθηθούν αλλού, θα επιστρέφουν αυτόματα στην INPUT. Αυτό δεν αποτελεί προϊόν κάποιας δικής μας ρύθμισης αλλά προκύπτει από τον τρόπο λειτουργίας του ίδιου του iptables. Έτσι, τα πακέτα που διέρχονται με επιτυχία από την αλυσίδα inspect θα ελέγχονται και από τον τελευταίο κανόνα της αλυσίδας INPUT (γραμμή 08). Αυτός ο κανόνας εξασφαλίζει ότι τα πακέτα που ανήκουν σε κάποια εγκαθιδρυμένη σύνδεση ή σχετίζονται με κάποια σύνδεση που ξεκίνησε από το σύστημά μας, θα γίνονται αποδεκτά. Όλα τα υπόλοιπα θα ακολουθούν τη γενική πολιτική της αλυσίδας INPUT και θα απορρίπτονται!

Στις γραμμές 09 έως 11 βλέπουμε τους κανόνες για την αλυσίδα OUTPUT. Από αυτή την αλυσίδα διέρχονται τα εξερχόμενα πακέτα και υπήρξαμε κάπως πιο ελαστικοί. Καταρχάς, επιλέξαμε σαν γενική πολιτική την αποδοχή (ACCEPT) των πακέτων και άρα την ελεύθερη διέλευσή τους. Ο επόμενος κανόνας απορρίπτει τα πακέτα που παρουσιάζουν κάποιο πρόβλημα (π.χ., δεν αποτελούν το ξεκίνημα μιας σύνδεσης αλλά δεν σχετίζονται ούτε και με κάποια εγκαθιδρυμένη). Στη γραμμή 11 πραγματοποιείται και πάλι μια προώθηση στην αλυσίδα inspect. Αναρωτιέστε γιατί κάναμε κάτι

τέτοιο; Το δικό μας σύστημα δεν θα θέλαμε να συμμετάσχει ποτέ σε κάποια επίθεση DoS. Επομένως, στα εξερχόμενα πακέτα θα εφαρμόσουμε όλους τους ελέγχους που αφορούν στις επιθέσεις DoS.

Το script συνεχίζει με τη ρύθμιση της αλυσίδας log_n_drop. Όπως βλέπετε, οι δύο σχετικοί κανόνες κάνουν αυτό που περιγράφει και το όνομα της αλυσίδας: Επιβάλλουν την καταγραφή των διερχόμενων πακέτων και τελικά την απόρριψή τους.

Στις τελευταίες γραμμές του script (14 έως 22) εκδηλώνεται η λανθασμένη αντίληψή μας για το πώς μπορεί να μπλοκάρει κανείς μια επίθεση DoS. Παρ' όλα αυτά οι συγκεκριμένες γραμμές έχουν αρκετό ενδιαφέρον, αφού μάς επέτρεψαν να εντοπίσουμε ένα ύπουλο λογικό σφάλμα και βοήθησαν τη σκέψη μας να προχωρήσει σε επόμενο level ;) Σε αυτές τις γραμμές ορίζονται οι κανόνες της αλυσίδας inspect, οι οποίοι προσπαθούν να μπλοκάρουν τις επιθέσεις DoS. Ε, λοιπόν, η (παραπλανητική) αλήθεια είναι ότι το πετυχαίνουν! Εφαρμόζοντας αυτούς τους κανόνες, το iptables ελέγχει το ρυθμό με τον οποίο καταφθάνουν πακέτα UDP προς τη θύρα 80 όπως και τα πακέτα TCP, στα οποία είναι ενεργοποιημένα τα PSH και ACK flags. Αν ο ρυθμός διέλευσης των σχετικών πακέτων ξεπεράσει τα όρια που έχουμε θέσει[4], το firewall αρχίζει την προώθησή τους στην αλυσίδα log_n_drop. Βλέπετε το λάθος μας; Όπως και να 'χει, μπορείτε να διαβάσετε τη συνέχεια της μικρής μας περιπέτειας στο άρθρο που ξεκινά από τη σελίδα 60.



Στις επιθέσεις HTTP flood τα πράγματα είναι πιο απλά. Τα αιτήματα που στέλνει το LOIC περιέχουν ένα χαρακτηριστικό λάθος, που καθιστά τον εντοπισμό τους παιχνιδάκι!

4. Ένα πακέτο ανά δευτερόλεπτο! Πάνω από αυτό το όριο (--limit), μόνο τα πρώτα οκτώ πακέτα θα γίνουν αποδεκτά (--limit-burst) και θα προωθηθούν. Τα υπόλοιπα θα αγνοηθούν και θα εξεταστούν από τον επόμενο κανόνα...



Πλήρη άρθρα που **συμπληρώνουν** το περιοδικό. Τα **video tutorials**. Άρθρα που μας ήρθε να γράψουμε, έτσι, απλά. Τα προκλητικά **mind hacks**. Τα βίντεο από τα προηγούμενα live streams. Ολόκληρα blasts from the past. Τα πάντα ενδιαφέροντα **fora**. Το σημείο συνάντησης για τους φίλους του περιοδικού.

deltahacker.gr

Αληθινό, διασκεδαστικό hacking **για όλους!**

Skill: Beginner
Tags: Syncthing, BitTorrent Sync, file synchronization

The king is dead, long live the king!

Syncthing

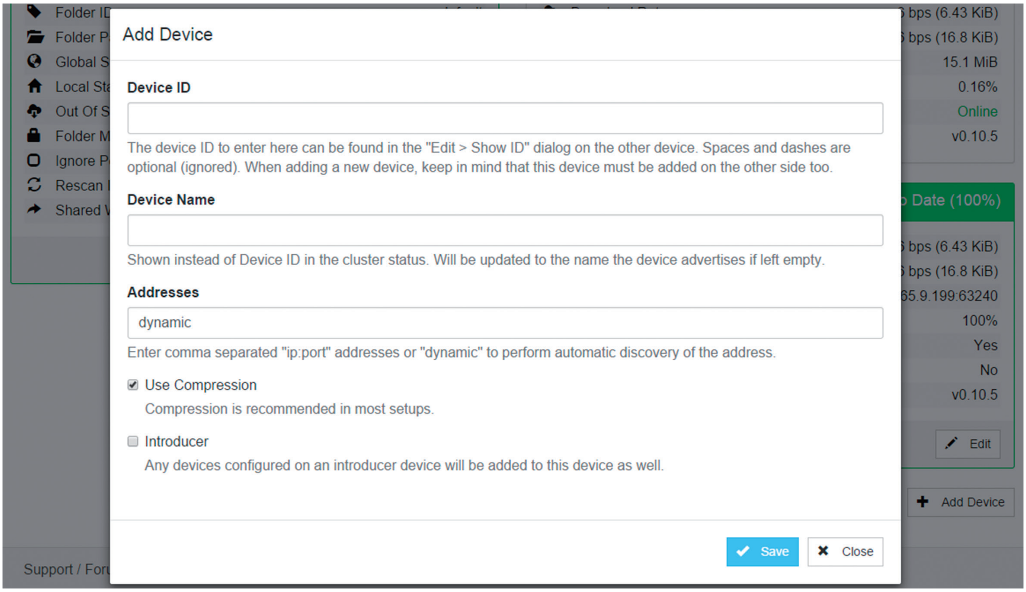
by multiPetros

Syncthing: The king is dead, long live the king!

Ο κατανεμημένος συγχρονισμός αρχείων μεταξύ υπολογιστών και συσκευών είναι η απάντηση των χρηστών στις όλο και πιο αυστηρές πολιτικές ελέγχου, που ακολουθούν διάφορες δημοφιλείς υπηρεσίες cloud storage. Φυσικά, οι υλοποιήσεις που στηρίζονται σε προγράμματα κλειστού κώδικα δεν μας επιτρέπουν να ησυχάσουμε. Κάπως έτσι στραφήκαμε στο Syncthing και στη θαλπωρή της κοινότητας του Ελεύθερου Λογισμικού.

Το BitTorrent Sync, το αποκεντρωμένο σύστημα συγχρονισμού και κοινής χρήσης αρχείων, μάς έχει απασχολήσει αρκετές φορές στο παρελθόν. Η αλήθεια είναι άλλωστε πως έχει καταφέρει να κερδίσει μια περίοπτη θέση στις καρδιές –αλλά και στα μηχανήματα– των απανταχού Parabingerz ;) (Σ.τ.Ε. Με άκουσαν στην πολυκατοικία, από τα γέλια που έβαλα.) Παρ' όλα αυτά, θα ήταν ψέμα να πούμε ότι δεν μας προβληματίζει η "κλειστή φύση" του κώδικά του. Αυτό δεν σημαίνει απαραίτητα κάτι κακό για την ποιότητα του προγράμματος ή την προστασία της ιδιωτικότητάς μας. Οι δημιουργοί του BitTorrent Sync ισχυρίζονται ότι το έργο τους διασφαλίζει πλήρως τα δικαιώματά μας και δεν υπάρχει κάποιος πρόδηλος λόγος να τους αμφισβητούμε. Ωστόσο, θα νιώθαμε πιο σίγουροι αν μπορούσαμε να ρίξουμε μια ματιά στον πηγαίο κώδικα του προγράμματος.

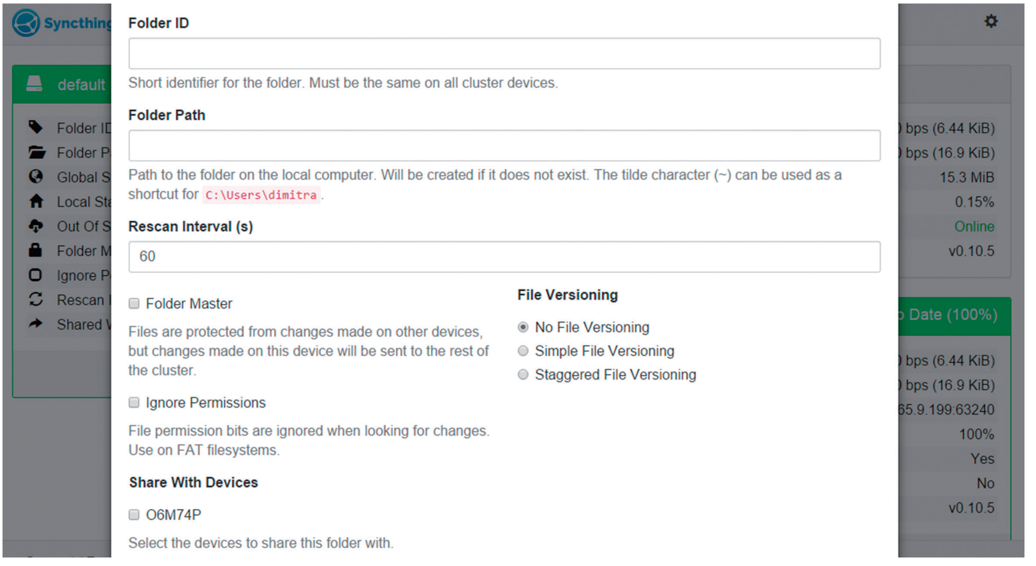
Την ανησυχία αυτή συμμερίζονται πολλοί άνθρωποι ανά τον κόσμο. Ένας απ' αυτούς, ο Jakob Borg, δεν περιορίστηκε στο να αισθάνεται άβολα και ξεκίνησε την ανάπτυξη μιας εναλλακτικής λύσης, ανοιχτού κώδικα. Κάπως έτσι γεννήθηκε το Syncthing! Με τη βοήθεια της κοινότητας προγραμματιστών ανοιχτού λογισμικού, η εφαρμογή εξελίχθηκε και ωρίμασε σε τέτοιο βαθμό που η τελευταία έκδοση του BitTorrent Sync δανείστηκε αρκετά στοιχεία της. Από τον περασμένο Οκτώβριο το Syncthing εντάχθηκε στο ευρύτερο σχέδιο ind.ie[1] και ξεκίνησε η διαδικασία μετονομασίας του σε Pulse. Βέβαια, αυτή η διαδικασία δεν έχει ολοκληρωθεί ακόμα και το Syncthing αναφέρεται από τον περισσότερο κόσμο με το αρχικό του όνομα.



Η διασύνδεση με μια νέα συσκευή επιτυγχάνεται πανεύκολα από το web interface του Syncthing. Το μόνο που χρειαζόμαστε είναι το ID του συγκεκριμένου μηχανήματος.

Το Syncthing είναι γραμμένο στη γλώσσα προγραμματισμού Go, που έχει αναπτύξει η Google. Το συντακτικό της εν λόγω γλώσσας έχει εμφανείς επιρροές από την Pascal και την Python, ενώ γι' αυτήν προσφέρονται compilers για πάρα πολλά λειτουργικά συστήματα. Έτσι, το Syncthing μπορεί να τρέξει σε όλα τα δημοφιλή λειτουργικά

1. Το ind.ie έχει σκοπό την ανάπτυξη και χρηματοδότηση μιας ομπρέλας εφαρμογών ανεξάρτητων δημιουργών, με σαφή προσανατολισμό την εξασφάλιση της ιδιωτικότητας και της ασφαλείας των δεδομένων των χρηστών.



Η προσθήκη ενός φακέλου προς συγχρονισμό ακολουθεί παρόμοια λογική, με εκείνη της διασύνδεσης συστημάτων. Ωστόσο, οι έξτρα "κινήσεις" που απαιτούνται από μέρους μας αξίζουν τον κόπο με το παραπάνω! Μην ξεχάσετε να επιλέξετε με ποιες από τις συνδεδεμένες συσκευές θα είναι δυνατός ο συγχρονισμός.

συστήματα (Windows, Linux, BSD, OSX και Android), καθώς και σε κάποιες (όχι και τόσο) εξωτικές εκδοχές του Unix που προορίζονται για επεξεργαστές ARM (v5, v6 και v7). Όπως καταλαβαίνετε αυτό το τελευταίο μάς φτιάχνει ιδιαιτέρως τη διάθεση, καθώς συνεπάγεται την απρόσκοπτη λειτουργία του Syncthing στο αγαπημένο μας Raspberry Pi! Να σημειώσουμε ότι εκτός από την 32μπιτη εκδοχή του προγράμματος, προσφέρεται και μια 64μπιτη.

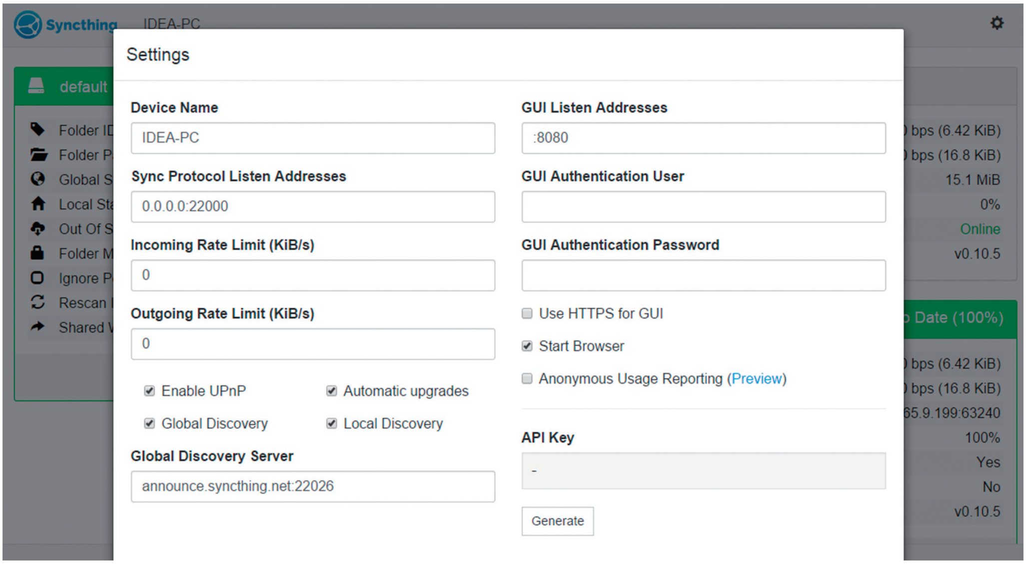
Ευκολία και ομοιομορφία

Η εγκατάσταση του Syncthing είναι πανεύκολη. Αρκεί να κατεβάσουμε το συμπιεσμένο πακέτο που αντιστοιχεί στο σύστημά μας, από τη διεύθυνση

`https://github.com/syncthing/syncthing/releases/latest`

Μετά την αποσυμπίεση μπορούμε απλά να τρέξουμε το εκτελέσιμο Syncthing. Κάνοντας κάτι τέτοιο σε κάποιο περιβάλλον γραφικών, θα εμφανιστεί ένα παράθυρο κονσόλας στο οποίο προβάλλονται πληροφορίες για την κατάσταση και τη λειτουργία του προγράμματος. Πρόκειται για την τυπική έξοδο της εφαρμογής (standard output). Τα ίδια μηνύματα εμφανίζονται και στην κονσόλα μας, στην περίπτωση που εκτελέσουμε το πρόγραμμα από τη γραμμή εντολών. Όπως αντιλαμβάνεστε, λοιπόν, στα συστήματα Unix –και για να μη δεσμεύουμε ένα τερματικό–, το screen θα αποτελεί πολύτιμο βοήθημα για την εκτέλεση του Syncthing. Αντίθετα, στα Windows θα πρέπει να υποστούμε την παρουσία ενός παραθύρου κονσόλας ή να καταφύγουμε σε μια custom λύση που θα παρουσιάσουμε αργότερα.

Το περιβάλλον εργασίας του Syncthing είναι απόλυτα ομοιόμορφο σε όλα τα λειτουργικά συστήματα. Αυτό συμβαίνει γιατί η διαχείριση του προγράμματος επιτυγχάνεται από ένα web interface, μέσω ενός web server που σηκώνεται αυτόματα κατά την εκκίνηση της εφαρμογής. Παρεμπιπτόντως, αυτό είναι το χαρακτηριστικό που



Το παράθυρο των γενικών ρυθμίσεων του Syncthing. Απλό, πλήρες και άκρως κατατοπιστικό!

δανείστηκε το BitTorrent Sync. Το παράθυρο του εν λόγω προγράμματος αποτελεί έναν μικρό web browser, που προβάλλει το σχετικό web interface. Επιστρέφοντας στα δικά μας, αξίζει να σημειώσουμε ότι το περιβάλλον του Syncthing στηρίζεται στο Bootstrap Framework, με αποτέλεσμα να είναι όμορφο, λειτουργικό και responsive. Ο web server που ενσωματώνει το Syncthing ακούει εξ ορισμού στη θύρα 8080, αλλά αυτό αλλάζει πανεύκολα από το αρχείο ρυθμίσεων του προγράμματος (~/.config/syncthing/config.xml στα συστήματα Unix και AppData%\Local\Syncthing\config.xml στα Windows). Αφού εντοπίσουμε και ανοίξουμε το αρχείο, αρκεί να μεταβούμε στην ακόλουθη ενότητα και να αλλάξουμε τη δικτυακή θύρα κατά το δοκούν:

```
<gui enabled="true" tls="false">
<address>127.0.0.1:8080</address>
</gui>
```

Από την ίδια περιοχή του αρχείου ρυθμίσεων μπορούμε να ενεργοποιήσουμε και την κρυπτογραφημένη σύνδεση μέσω https, ορίζοντας την τιμή της ιδιότητας tls σε true. Επιπρόσθετα, μπορούμε να ορίσουμε και έναν λογαριασμό χρήστη, εισάγοντας μέσα στο tag <gui> τις ακόλουθες γραμμές:

```
<user>a_user_name</user>
<password>a_secure_password</password>
```

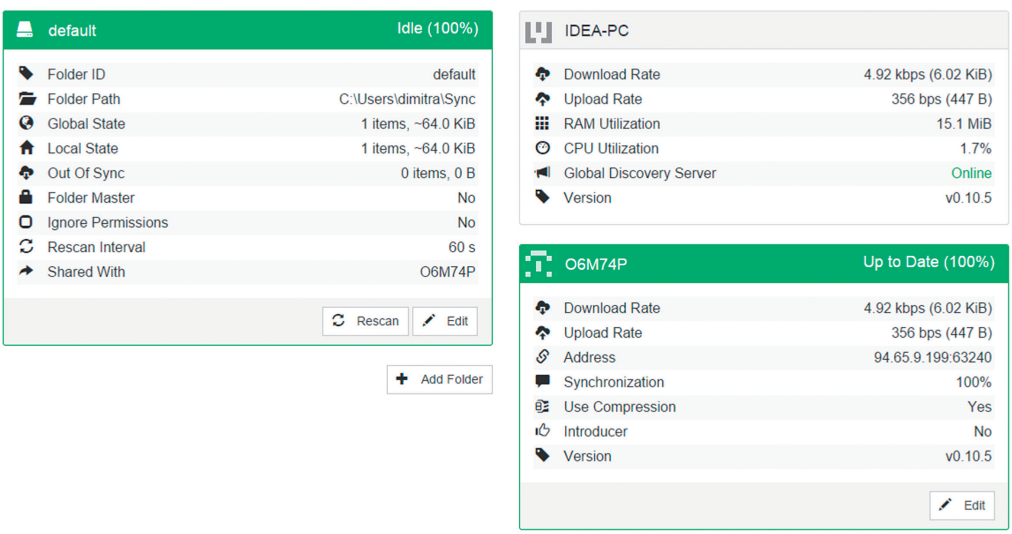
Σημειώστε ότι το όνομα χρήστη και το αντίστοιχο συνθηματικό εισάγονται σε μορφή plain text. Αν σκέπτεστε ότι η αποθήκευση του password σε αυτή τη μορφή δεν είναι καθόλου ασφαλής, έχετε απόλυτο δίκιο. Ακριβώς γι' αυτό, με την πρώτη επανεκτέλεση του προγράμματος το string του συνθηματικού θα αντικατασταθεί αυτόματα από ένα κρυπτογραφημένο hash. Τέλος, από το tag <gui> μπορούμε και να απενεργοποιήσουμε το web interface πλήρως, ορίζοντας την τιμή της ιδιότητας enabled σε false.

Παραμετροποίηση με άνεση

Από το ίδιο αρχείο ρυθμίσεων μπορούμε να τροποποιήσουμε πολλές ακόμα πλευρές της λειτουργίας του προγράμματος. Ωστόσο, μετά τις βασικές ρυθμίσεις που είδαμε προηγουμένως, μπορούμε τώρα να καταφύγουμε στην ευκολία του web interface. Οι επιλογές του προγράμματος εμφανίζονται κάνοντας κλικ στο εικονίδιο με το γρανάζι (στην πάνω δεξιά άκρη) κι επιλέγοντας το Settings, από το μενού που αναδύεται.

Μια σημαντική ρύθμιση, πέρα από αυτή της δικτυακής θύρας, είναι η διεύθυνση IP. Αν παραλείψουμε να ορίσουμε κάποια τιμή στο σχετικό πεδίο, η πρόσβαση στο web interface θα επιτρέπεται από όλα τα μηχανήματα του δικτύου. Έτσι, τις περισσότερες φορές είναι προτιμότερο να αφήνουμε την προεπιλογή (localhost ή 127.0.0.1), ώστε η πρόσβαση να επιτρέπεται μόνο από το μηχάνημα στο οποίο εκτελείται το Syncthing. Μερικές ακόμα ενδιαφέρουσες ρυθμίσεις αφορούν στον καθορισμό των ορίων για το bandwidth αποστολής και λήψης δεδομένων, στο αν θα εμφανίζεται αυτόματα το web interface όταν ξεκινάμε το πρόγραμμα, όπως επίσης στο αν και στο πώς θα αναζητούνται συσκευές που τρέχουν το Syncthing.

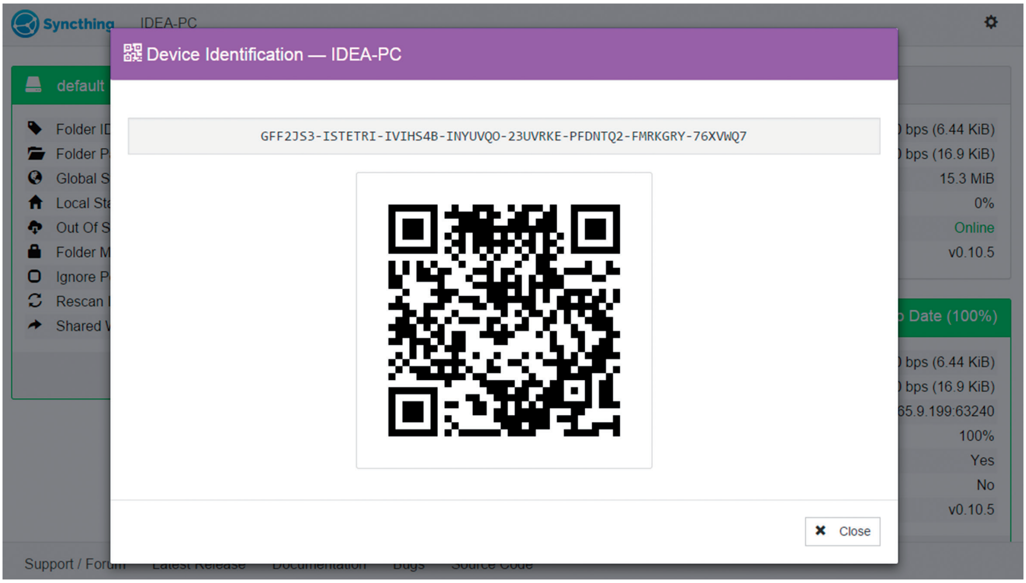
Αν οι υπολογιστές μεταξύ των οποίων θέλουμε να συγχρονίζουμε αρχεία βρίσκονται στο ίδιο τοπικό δίκτυο, μπορούμε για λόγους ασφάλειας να απενεργοποιήσουμε το Global Discovery. Επιπρόσθετα, μπορούμε να απενεργοποιήσουμε και το Local Discovery, με την προϋπόθεση ότι θα δώσουμε χειροκίνητα τις διευθύνσεις IP των μηχανημάτων που συγχρονίζονται.



Στην κεντρική περιοχή του web interface δεσπόζουν οι πληροφορίες για το ίδιο το πρόγραμμα, τις διασυνδεδεμένες συσκευές και τους φακέλους που συγχρονίζονται.

Διασύνδεση συστημάτων

Η φιλοσοφία συγχρονισμού των αρχείων με το Syncthing διαφέρει από εκείνη του BitTorrent Sync. Με το Syncthing δεν συνδεόμαστε απευθείας σε κοινόχρηστους πόρους, αλλά πραγματοποιούμε συνδέσεις μεταξύ συσκευών. Για όσους είναι εξοικειωμένοι με το BitTorrent Sync, η παραπάνω διαφορά μπορεί να περιγραφεί και ως εξής:



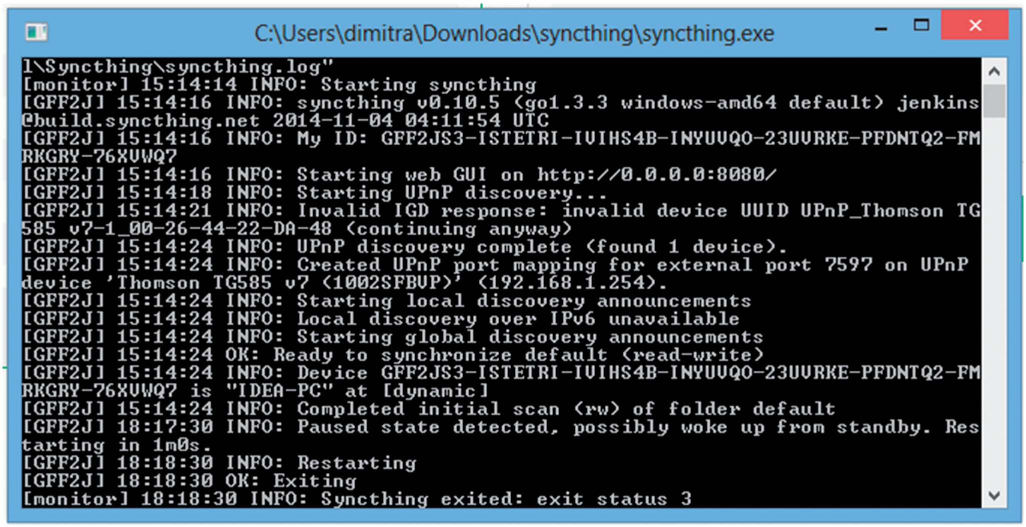
Το ID είναι το βασικό κλειδί για τη διασύνδεση συσκευών που τρέχουν το Syncthing. Το QR Code εμφανίζεται για την περίπτωση που θέλουμε να μεταφέρουμε εύκολα ένα ID σε μια συσκευή με το Android.

Αντί να δίνουμε τα μακροσκελή ID των φακέλων που θέλουμε να συγχρονίσουμε, δίνουμε τα ID των διαφορετικών υπολογιστών που θέλουμε να συνδέσουμε. Στη συνέχεια, έχουμε τη δυνατότητα να ορίσουμε με ακρίβεια ποιοι υπολογιστές (από τους διασυνδεδεμένους) θα συγχρονίζουν ποιους φακέλους. Ίσως αυτή η περιγραφή να σας μπέρδεψε, αλλά θα ρίξουμε φως σε κάθε σκοτεινό σημείο με τη βοήθεια ενός απλού παραδείγματος. Ας δούμε τη διαδικασία για το συγχρονισμό δύο καταλόγων, δύο διαφορετικών υπολογιστών.

Προφανώς, ξεκινάμε εγκαθιστώντας το Syncthing στα δύο μηχανήματα. Το επόμενο βήμα είναι να συνδέσουμε τα συστήματα μεταξύ τους. Για το σκοπό αυτό θα χρειαστούμε το ID του καθενός, που εμφανίζεται αν επιλέξουμε το Show ID στο μενού ρυθμίσεων (πίσω από το εικονίδιο με το γρανάζι). Εφόσον έχουμε σημειώσει τα δύο ID, συνδεόμαστε στο web interface του ενός συστήματος και πατάμε το κουμπί Add Device. Στο παράθυρο που εμφανίζεται δίνουμε το ID του άλλου συστήματος και, προαιρετικά, δηλώνουμε ένα όνομα ώστε να θυμόμαστε εύκολα για ποιο μηχάνημα πρόκειται. Στο πεδίο Addresses δίνουμε τη διεύθυνση IP του άλλου υπολογιστή ή, στην περίπτωση που από τις ρυθμίσεις έχουμε επιτρέψει την αυτόματη αναζήτηση συσκευών, αφήνουμε την προεπιλεγμένη τιμή (dynamic). Η χρησιμότητα του Use Compression είναι προφανής και δεν χρειάζεται να πούμε κάτι. Τέλος, η επιλογή Introducer κάνει κάτι πολύ χρήσιμο. Φανταστείτε την περίπτωση που έχετε συνδέσει πολλούς υπολογιστές με το Syncthing και κάποια στιγμή αποφασίζετε να προσθέσετε ένα νέο μηχάνημα σε αυτό το "δίκτυο συγχρονισμού". Αν έχετε ενεργοποιήσει τη συγκεκριμένη επιλογή σε κάθε μηχάνημα, μόλις προσθέσετε το νέο σύστημα σε ένα μέλος του δικτύου θα προστεθούν αυτόματα και τα υπόλοιπα.

Μετά από το παραπάνω πατάμε το κουμπί Save και η διαδικασία σύνδεσης των δύο συστημάτων θα έχει ολοκληρωθεί κατά το ήμισυ. Το όνομα του συστήματος που προσθέσαμε θα εμφανιστεί στην κεντρική περιοχή του web interface σε ένα κουτά-

κι με μωβ χρώμα. Το συγκεκριμένο χρώμα υποδεικνύει ότι τα δύο συστήματα δεν έχουν συνδεθεί ακόμα. Για να ολοκληρωθεί η διασύνδεσή τους πρέπει να επαναλάβουμε την παραπάνω διαδικασία και στο δεύτερο μηχάνημα, δίνοντας το ID του πρώτου. Αμέσως μετά θα μπορούμε να περάσουμε στο επόμενο βήμα, κατά το οποίο επιλέγουμε τους φακέλους που θα συγχρονίζονται.



Το παράθυρο της γραμμής εντολών είναι πάντα ορατό κατά την εκτέλεση του Syncthing στα Windows. Μην νομίζετε ότι θα το ανεχτούμε αυτό ;)

Κατάλογοι προς συγχρονισμό

Αρχικά πατάμε το κουμπί Add Folder στο web interface του ενός μηχανήματος. Στο παράθυρο που εμφανίζεται δίνουμε τη διαδρομή του τοπικού φακέλου που θέλουμε να συγχρονίζεται, καθώς και ένα περιγραφικό όνομα που θα λειτουργεί σαν ID του συγκεκριμένου φακέλου στο δίκτυο. Ακολουθεί ο ορισμός του μεγέθους Rescan Interval, που καθορίζει το χρονικό διάστημα ανά το οποίο θα ελέγχονται τα αρχεία του φακέλου για αλλαγές. Η προεπιλεγμένη τιμή των 60 δευτερολέπτων είναι αρκετά ικανοποιητική. Ωστόσο, στην περίπτωση που ο συγκεκριμένος φάκελος χρησιμοποιείται σπάνια, μπορούμε να δώσουμε μια ακόμα μεγαλύτερη τιμή για να περιορίσουμε την επιβάρυνση του συστήματος. Λίγο χαμηλότερα και στα δεξιά συναντάμε μια ιδιαίτερα χρήσιμη ρύθμιση, η οποία αφορά στη δυνατότητα διατήρησης πολλαπλών εκδόσεων των αρχείων μας. Ουσιαστικά, μιλάμε για το γνωστό file versioning. Αν ενεργοποιήσουμε το λεγόμενο Simple File Versioning, μπορούμε να προσδιορίσουμε τον επιθυμητό αριθμό εκδόσεων που θα διατηρούνται για κάθε αρχείο. Οι παλιότερες εκδόσεις αποθηκεύονται σε έναν κατάλογο που δημιουργείται αυτόματα και έχει το όνομα ".stversions". Η επιλογή του λεγόμενου Staggered File Versioning διατηρεί αυτόματα πολλές εκδόσεις των αρχείων μας, ανάλογα με την παλαιότητά τους. Συγκεκριμένα, για τις αλλαγές της τελευταίας μίας ώρας διατηρούνται αντίγραφα ανά 30 δευτερόλεπτα, από την τελευταία μία ημέρα διατηρούνται αντίγραφα ανά μία ώρα, για τον πρώτο μήνα ένα αντίγραφο ανά ημέρα και πάει λέγοντας. Όπως αντιλαμβάνεστε, το File Versioning είναι ιδιαίτερα χρήσιμο για έναν φάκελο με αρχεία της δουλειάς μας, αλλά η χρήση του απαιτεί μια κάποια σύνεση, για να μην καταλήξουμε να γεμίσουμε τους δίσκους μας με άχρηστα αντίγραφα.

Στα αριστερά του File Versioning βρίσκονται μερικές ακόμα επιλογές για τον τρόπο συγχρονισμού του φακέλου. Παρόμοιες επιλογές προσφέρει και το BitTorrent Sync, ενώ το web interface περιλαμβάνει και μια σύντομη επεξήγηση της καθεμίας. Τέλος, στην ενότητα Share With Devices, εμφανίζεται μια λίστα με τις συνδεδεμένες συσκευές. Από εκεί μπορούμε να επιλέξουμε με ποια μηχανήματα θα συγχρονίζεται ο συγκεκριμένος φάκελος. Μετά κι από αυτό, πατάμε στο κουμπί Save και είμαστε *σχεδόν* έτοιμοι. Όπως κατά την αρχική διασύνδεση των υπολογιστών, έτσι και τώρα η παραπάνω διαδικασία πρέπει να επαναληφθεί και στο άλλο άκρο. Αφού συνδεθούμε στο web interface του άλλου συστήματος, πρέπει να πατήσουμε πάλι το κουμπί Add Folder. Αυτή τη φορά οφείλουμε να δώσουμε το ID που αντιστοιχίσαμε νωρίτερα στον κατάλογο του άλλου συστήματος, καθώς και να ορίσουμε τη θέση στην οποία θα αποθηκεύονται τα αρχεία του απομακρυσμένου καταλόγου!

Μετά κι από το δεύτερο βήμα που μόλις περιγράψαμε, θα ξεκινήσει ο συγχρονισμός του φακέλου μεταξύ των δύο μηχανημάτων. Ο συγκεκριμένος κατάλογος θα εμφανίζεται πλέον και σ' ένα πλαίσιο, στην κεντρική περιοχή του web interface των δύο συστημάτων. Στην κάτω πλευρά αυτού του πλαισίου υπάρχουν τα κουμπιά Rescan και Edit. Το πρώτο επιβάλει τον άμεσο έλεγχο του φακέλου για πιθανές αλλαγές στο περιεχόμενό του, ενώ το δεύτερο εμφανίζει τις ρυθμίσεις συγχρονισμού του φακέλου. Πέρα από όσα αναφέραμε προηγουμένως, στο κάτω μέρος του παραθύρου θα συναντήσουμε δύο πρόσθετα κουμπιά: Το Delete και το Ignore Patterns. Το πρώτο έχει μάλλον παραπλανητικό όνομα και το μόνο που κάνει είναι να καταργεί το συγχρονισμό του συγκεκριμένου καταλόγου, χωρίς να τον διαγράφει από το δίσκο. Το δεύτερο επιτρέπει να ορίσουμε μοτίβα ονομάτων, για τα αρχεία που *δεν* θέλουμε να συγχρονίζονται.

Σημειώστε ότι κατά τη διάρκεια των εργασιών μας με το Syncthing θα εμφανιστεί αρκετές φορές ένα πλαίσιο με την προτροπή για επανεκκίνηση του προγράμματος. Αυτό δεν πρέπει να μας ανησυχεί καθόλου, αφού η σχετική διαδικασία πραγματοποιείται ταχύτατα και δεν επηρεάζει το υπόλοιπο σύστημα.

Ψεγάδι, αλλά αντιμετωπίσιμο

Το μόνο αρνητικό που θα μπορούσαμε να καταλογίσουμε στο Syncthing σχετίζεται με την υποχρεωτική εμφάνιση του παραθύρου της γραμμής εντολών, όταν τρέχουμε το πρόγραμμα στα Windows. Για τους 5-6 κατοίκους του πλανήτη που δεν γνωρίζουν τα χούγια των απανταχού Parabingerz, πρέπει να πούμε ότι είναι αδύνατο να μείνουμε με σταυρωμένα χέρια μπροστά σ' ένα τέτοιο θέμα. Εφόσον η ομάδα ανάπτυξης του Syncthing δεν έχει δώσει λύση, αποφασίσαμε να φτιάξουμε τη δική μας. Αξιοποιώντας το Windows API και τη γλώσσα C, κατασκευάσαμε μια παραθυρική εφαρμογή ...χωρίς παράθυρο! Η αποστολή του προγράμματός μας είναι να εκτελεί στο παρασκήνιο το Syncthing, επιτυγχάνοντας έτσι την απόκρυψη του ενοχλητικού παραθύρου. Μάλιστα, το πρόγραμμά μας τερματίζεται αυτόματα και αφήνει το Syncthing να τρέχει ανενόχλητο και πάνω απ' όλα "αθόρυβα". Ακολουθεί ο κώδικας του προγράμματος:

```
/* Syncthing Silent Runner - v1.0
 * Run Syncthing at MS Windows without the console's window
 * Copyright (c) 2014, Petros Kyladitis <http:\\www.multipetros.gr\>
```

```
04  * This is free software distributed under the terms of the GNU GPL v3
05  */
06
07 #include <windows.h>
08
09 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR szCmdLine, int iCmdShow){
10   int execResult = WinExec("syncthing.exe" , SW_HIDE) ;
11   if(execResult == ERROR_BAD_FORMAT || execResult == ERROR_FILE_NOT_FOUND || execResult == ERROR_PATH_NOT_FOUND){
12      MessageBoxA(NULL, "Syncthing executable didn\'t found. Please put the \'Syncthing Silent Runner\' in the same folder where the Syncthing executable is placed.", "Syncthing Silent Runner", MB_ICONERROR | MB_OK);
13   }
14   else if(execResult == 0){
15      MessageBoxA(NULL, "Out of memory or system resources. Close some windows or programs and try again.", "Syncthing Silent Runner", MB_ICONERROR | MB_OK);
16   }
17   exit(0);
18 }
```

Όλη η δουλειά πραγματοποιείται από τη συνάρτηση WinExec του Windows API, στη γραμμή 10. Όπως βλέπετε, η εν λόγω συνάρτηση δέχεται δύο παραμέτρους: Το πρόγραμμα που θέλουμε να εκτελεστεί, καθώς και τον τρόπο (mode) εκτέλεσης. Χρησιμοποιώντας τη σταθερά SW_HIDE, επιλέγουμε την εκκίνηση του προγράμματος με *κρυμμένο* το παράθυρο κι όχι απλά ελαχιστοποιημένο. Η WinExec επιστρέφει ορισμένες τιμές, ανάλογα με την επιτυχία ή όχι της εκτέλεσης του προγράμματος. Στις γραμμές 11 έως 16 ελέγχονται αυτές οι τιμές και εμφανίζονται στο χρήστη τα αντίστοιχα μηνύματα. Μπορείτε να κατεβάσετε το μικρό αλλά αποτελεσματικό προγραμματάκι μας από το http://bit.ly/dh038stsr. Σημειώστε ότι για τη σωστή λειτουργία του προγράμματός μας πρέπει να το έχετε τοποθετήσει στον ίδιο κατάλογο με το syncthing.exe.

Το Syncthing ήρθε για μείνει κι αυτό φανερώνεται τόσο από τον ταχύτατο ρυθμό ανάπτυξής του, όσο κι από το γεγονός ότι αποτελεί πρότυπο για τις ανταγωνιστικές εφαρμογές. Ειδικά με τη λύση που παρουσιάσαμε για τους χρήστες Windows, δεν υπάρχει καμία δικαιολογία για να κολυμπάτε στο θολά νερά του κλειστού κώδικα ;)

Skill: Beginner
Tags: Kali Linux, Maltego, penetration testing, social engineering



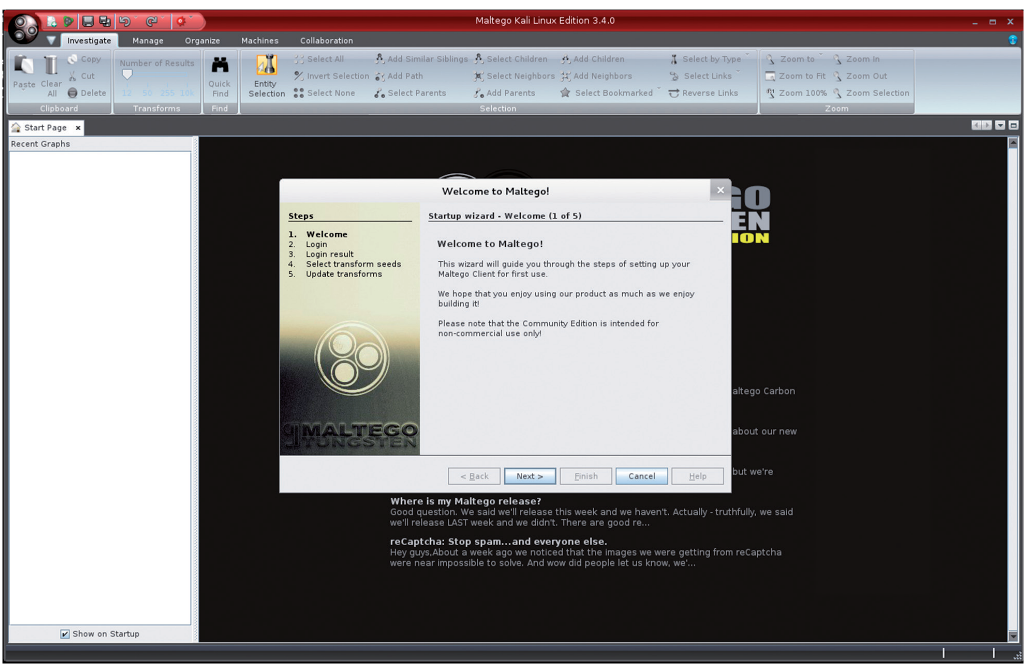
by Cr0wTom

Social engineering με το Maltego

Οι πλατφόρμες για pen-testing διαθέτουν δεκάδες εργαλεία που αξιοποιούν τις ευπάθειες των δικτυακών πρωτοκόλλων και υπηρεσιών. Λογικό, θα σκεφτεί κανείς. Ποιες άλλες αδυναμίες θα μπορούσαν να αξιοποιούν, δηλαδή; Εκείνες που απορρέουν από την αφέλεια ή την άγνοια του εκάστοτε χρήστη, ίσως; Ακριβώς. Υπάρχει μια πλατφόρμα που κάνει ακριβώς αυτό: Περιλαμβάνει εργαλεία για την εύκολη περισυλλογή προσωπικών δεδομένων, τα οποία θα μπορούσαν να αξιοποιηθούν σε επιθέσεις social engineering!

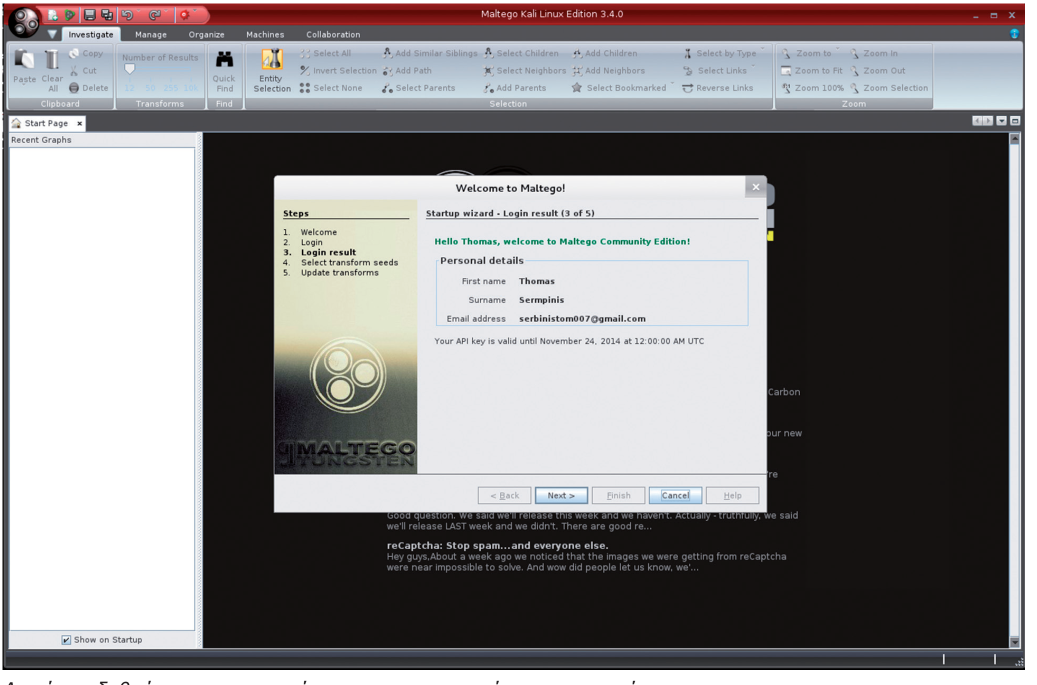
Η καθημερινότητα ενός pen-tester, είτε πρόκειται για επαγγελματία είτε για απλό χομπίστα, περιλαμβάνει την αδιάκοπη αναζήτηση –γνωστών και μη– ευπαθειών στα συστήματα που μας περιβάλλουν. Τις περισσότερες φορές, βέβαια, πρόκειται για γνωστές ευπάθειες που ξεφυτρώνουν διαρκώς, αν κι έχουν μελετηθεί σε βάθος προ πολλού. Έτσι, η δουλειά του pen-tester περιλαμβάνει την επανάληψη αρκετών χρονοβόρων εργασιών, που τσεκάρουν την ύπαρξη "κλασικών" ευπαθειών. Ακριβώς γι' αυτό το λόγο υπάρχουν πλατφόρμες όπως το γνωστό (κι αγαπημένο μας) Metasploit, που αυτοματοποιούν όλο το ανιαρό κομμάτι της δουλειάς. Σε αυτό το άρθρο θα ασχοληθούμε με μια πλατφόρμα του είδους, η οποία ωστόσο δεν αφορά στις ευπάθειες με τις οποίες συνήθως ασχολούμαστε. Ο λόγος γίνεται για το Maltego, που ενσωματώνει ένα σύνολο ισχυρών εργαλείων για τον εντοπισμό εν δυνάμει χρήσιμων πληροφοριών.

Το Maltego χρησιμοποιεί τη μέθοδο OSINT (Open-Source Intelligence, http://en.wikipedia.org/wiki/Open-source_intelligence) και συλλέγει πληροφορίες που βρίσκονται διάσπαρτες σε κοινωνικά δίκτυα ή και σε άλλες δικτυακές υπηρεσίες. Αναφερόμαστε σε προσωπικά δεδομένα που δημοσιεύτηκαν είτε από αφέλεια είτε ακούσια, όπως, είναι οι λογαριασμοί email, τα URL, οι "φίλοι" στα κοινωνικά δίκτυα και τα αρχεία (εικόνες, τραγούδια, βίντεο κ.λπ.) που έχει προσπελάσει κάποιος χρήστης. Στα δεδομένα που συλλέγει το Maltego μπορούμε να προσθέσουμε και δικές μας εγγραφές, με στοιχεία που γνωρίζουμε ήδη για το χρήστη-στόχο. Όλα αυτά τα δεδομένα παρουσιάζονται με έναν εξαιρετικά έξυπνο τρόπο, που μας επιτρέπει να διαχειριστούμε το μεγάλο τους όγκο. Το Maltego ενσωματώνει κι ένα εργαλείο που επιτρέπει σε απομακρυσμένους χρήστες να ανταλλάσσουν τα στοιχεία που έχουν συγκεντρώσει. Επιπρόσθετα, επιτρέπει τη συνεργασία σε πραγματικό χρόνο, μέσω του πρωτοκόλλου XMPP.



Τη πρώτη φορά που θα εκκινήσουμε το Maltego θα εμφανιστεί ένα παράθυρο για την εισαγωγή των στοιχείων του λογαριασμού μας στους server της Paterva. Αν δεν έχουμε λογαριασμό θα μπορέσουμε να δημιουργήσουμε έναν δωρεάν!

Πίσω από το Maltego βρίσκεται η εταιρία Paterva, η οποία έχει δημιουργήσει δύο εκδοχές της πλατφόρμας: Την εμπορική, που προορίζεται για χρήση στα εταιρικά δίκτυα μεγάλων επιχειρήσεων και συνοδεύεται από ένα τσουχτερό κόστος, όπως επίσης και την αποκαλούμενη Community, που διατίθεται δωρεάν για προσωπική χρήση. Σημειώστε ότι το Maltego είναι γραμμένο σε Java και μπορεί να τρέξει σε οποιοδήποτε σύστημα έχουμε εγκαταστήσει το JVM. Μπορείτε να κατεβάσετε τη δωρεάν εκδοχή της πλατφόρμας από εδώ `www.paterva.com/web6/products/download2.php` ή να χρησιμοποιήσετε το Kali Linux, στο οποίο βρίσκεται εγκατεστημένη εξ ορισμού. Εμείς κάναμε το δεύτερο.



Αφού συνδεθούμε με τα μηχανήματα της κατασκευάστριας εταιρείας, το πρόγραμμα θα πραγματοποιήσει ένα είδος συγχρονισμού και θα ενημερώσει τα εγκατεστημένα transforms (κάτι σαν extensions).

Συλλογή πληροφοριών

Το Maltego βρίσκεται στο μενού Kali Linux και στη θέση Information Gathering → OSINT Analysis. Μπορούμε να το τρέξουμε και από την κονσόλα, δίνοντας ένα απλό "maltego" (χωρίς τα εισαγωγικά). Την πρώτη φορά που θα το τρέξουμε θα μας ζητηθεί να κάνουμε εγγραφή του προϊόντος. Η διαδικασία περιλαμβάνει τη δημιουργία ενός λογαριασμού στους servers της Paterva και είναι δωρεάν. Ολοκληρώνοντας αυτή τη μικρή αγγαρεία, το πρόγραμμα θα συνδεθεί στα μηχανήματα της εταιρείας και θα ανανεώσει τα εγκατεστημένα transforms. Πρόκειται για αυτοτελή τμήματα κώδικα που συνοδεύουν το Maltego και τα οποία μπορείτε να φαντάζεστε ως extensions. Όπως και να 'χει, η ενημέρωσή τους είναι φυσιολογική και δεν χρειάζεται να σας προβληματίσει.

Μπορούμε να ξεκινήσουμε την εργασία μας συλλέγοντας μερικές πληροφορίες για το στόχο μας. Πηγαίνοντας στη καρτέλα Investigate και στο αριστερό τμήμα του

παραθύρου του προγράμματος, θα βρούμε μια λίστα με δύο υποκατηγορίες: Infrastructure και Personal. Η πρώτη (Infrastructure) περιλαμβάνει τις επιθέσεις που στηρίζονται σε τεχνικές ευπάθειες, ενώ η δεύτερη (Personal) περιέχει τις επιθέσεις που βασίζονται στον ανθρώπινο παράγοντα. Ξεκινώντας με την κατηγορία Infrastructure, βλέπουμε ότι η σχετική παλέτα περιέχει μια ευρεία γκάμα επιλογών. Η έρευνά μας μπορεί να βασιστεί σε ένα domain name, σε μια διεύθυνση IP, σε ένα MX Record ή σε κάποιο άλλο αναγνωριστικό στοιχείο για το σύστημα-στόχο. Αν προχωρήσουμε στην εξέταση ενός στόχου, τα αποτελέσματα θα εμφανιστούν στο κεντρικό παράθυρο του προγράμματος με τη μορφή ενός διαγράμματος. Στις επιθέσεις της κατηγορίας Personal και από την άποψη της εμφάνισης των αποτελεσμάτων, τα πράγματα δεν αλλάζουν πολύ. Οι επιθέσεις που θα βρούμε εδώ, όμως, είναι εντελώς διαφορετικής φύσης κι έχουν σαν στόχο χρήστες κι όχι συστήματα. Από εδώ μπορούμε να συλλέξουμε πληροφορίες από κοινωνικά δίκτυα, διευθύνσεις email, αρχεία που έχουν ανεβαστεί από το χρήστη-στόχο ή έχει αλληλεπιδράσει μαζί τους, διευθύνσεις URL και πολλά άλλα. Για παράδειγμα, αν επιλέξουμε την επίθεση Person, αρκεί να σύρουμε το σχετικό πλακίδιο-κουμπί στην κεντρική περιοχή του παραθύρου. Μόλις αφήσουμε το πλακίδιο, το πρόγραμμα θα μας ζητήσει να συμπληρώσουμε το όνομα του ανθρώπου για τον οποίο ψάχνουμε πληροφορίες. Στη συνέχεια, κάνοντας δεξί κλικ επάνω στο πλακίδιο, εμφανίζονται όλα τα διαθέσιμα transforms που αφορούν σε αυτόν τον τύπο επίθεσης. Ένα ενδιαφέρον transform για την επίθεση People είναι το λεγόμενο toFacebookaffiliation, που αναζητά προφίλ του Facebook τα οποία συνδέονται με το στόχο μας. Όπως αντιλαμβάνεστε, οι πληροφορίες που μπορούμε να συλλέξουμε με τις επιθέσεις Personal είναι ιδιαίτερα χρήσιμες όταν θέλουμε να κατασκευάσουμε προσαρμοσμένα word lists, ή όταν σκοπεύουμε να στραφούμε σε κάποια προχωρημένη τεχνική social engineering.

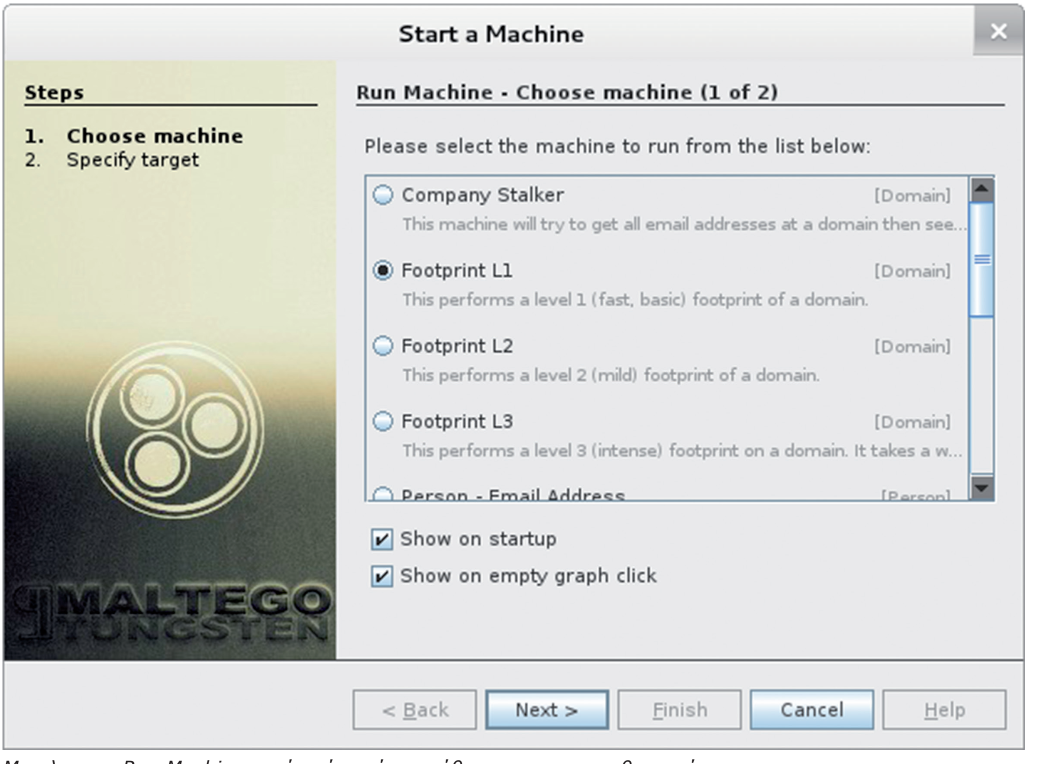


Πηγαίνοντας στην καρτέλα Machines του κεντρικού μενού βλέπουμε τέσσερις επιλογές: Την εκκίνηση μιας μηχανής, τον τερματισμό όλων των ενεργών μηχανών, τη δημιουργία νέας μηχανής και την επεξεργασία των διαθέσιμων μηχανών. Ξεκινώντας τη γνωριμία μας με το Maltego αρκεί να επιλέξουμε την εκκίνηση μιας (έτοιμης) μηχανής.

Γνωριμία με το περιβάλλον

Η εξοικείωση με το περιβάλλον του προγράμματος αποτελεί απαραίτητη προϋπόθεση για την αξιοποίηση των δυνατοτήτων του. Και φυσικά, ο καλύτερος τρόπος για να μάθει κανείς το περιβάλλον ενός προγράμματος είναι η ενασχόληση μαζί του. Εμείς ωστόσο θα πραγματοποιήσουμε μια σύντομη περιγραφή, για να περάσουμε στη συνέχεια στην παρουσίαση δύο σεναρίων χρήσης του προγράμματος. Το περιβάλλον του Maltego θα μπορούσε να χωριστεί σε πέντε περιοχές:

1. Στο επάνω μέρος του παραθύρου εμφανίζεται μια μπάρα επιλογών, από την οποία μπορούμε να πραγματοποιήσουμε ρυθμίσεις ή και να εκκινήσου-



Με κλικ στο Run Machine ανοίγει ένα νέο παράθυρο με τα προκαθορισμένα ήδη μηχανών. Στη πρώτη μας επίθεση επιλέγουμε το Footprint L1, που εκτελεί μια βασική έρευνα στοιχείων με βάση κάποιο Domain name. Στη δεύτερη επίθεση επιλέγουμε το Person-Email Address, που πραγματοποιεί έρευνες με βάση το όνομα του ανθρώπου–στόχου.

με κάποια διαδικασία. Η εν λόγω μπάρα περιλαμβάνει τις εξής καρτέλες: Investigate, Manage, View, Organize, Machines και Collaboration. Στην καρτέλα Investigate βρίσκονται οι λειτουργίες που αφορούν στην επεξεργασία των ευρημάτων μας. Η καρτέλα Manage επιτρέπει τη διαχείριση των επιθέσεων και των διαθέσιμων transforms. Στις καρτέλες View και Organize εμφανίζονται επιλογές που αφορούν στην εμφάνιση του προγράμματος και των δεδομένων που προκύπτουν από τις έρευνες. Από την καρτέλα Machines μπορούμε να ξεκινήσουμε μια νέα έρευνα, ενώ από την καρτέλα Collaboration μπορούμε να μοιραστούμε τα ευρήματά μας με άλλους ερευνητές, μέσω του πρωτοκόλλου XMPP.

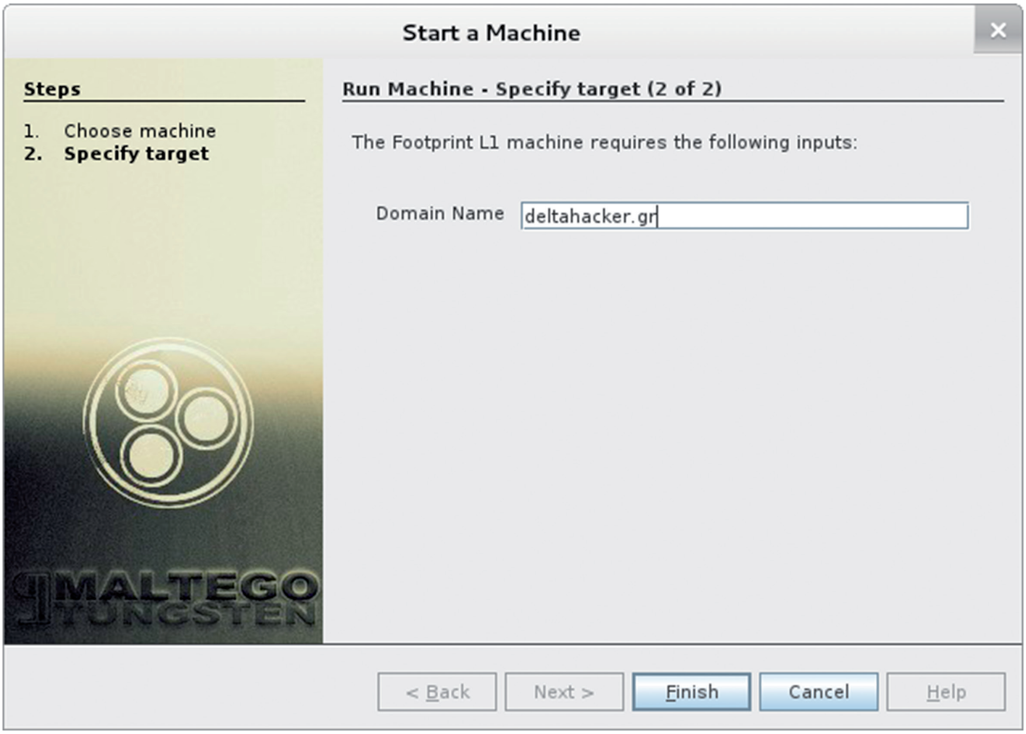
2. Η κεντρική περιοχή του παραθύρου περιλαμβάνει τα λεγόμενα Workspaces. Καθένα από αυτά αντιστοιχεί σε μια έρευνα και παρουσιάζει τα αντίστοιχα ευρήματα, με τη μορφή ενός διαγράμματος.

3. Ακριβώς κάτω από τα Workspaces βρίσκεται η περιοχή Output. Εκεί εμφανίζονται μηνύματα απλού κειμένου, που περιγράφουν την τρέχουσα δραστηριότητα του προγράμματος.

4. Στη δεξιά πλευρά του παραθύρου διακρίνονται τρία μικρά παράθυρα: Overview, Detail View, Property View. Το περιεχόμενο καθενός παραθύρου

είναι σχεδόν αυτονόητο. Το πρώτο προσφέρει τη γενική εποπτεία του Maltego και των ενεργών ερευνών, το δεύτερο εμφανίζει πληροφορίες για την επιλεγμένη έρευνα, ενώ το τρίτο προβάλλει τις ιδιότητες του στοιχείου που έχουμε επιλέξει από το διάγραμμα που εμφανίζεται στο ενεργό workspace.

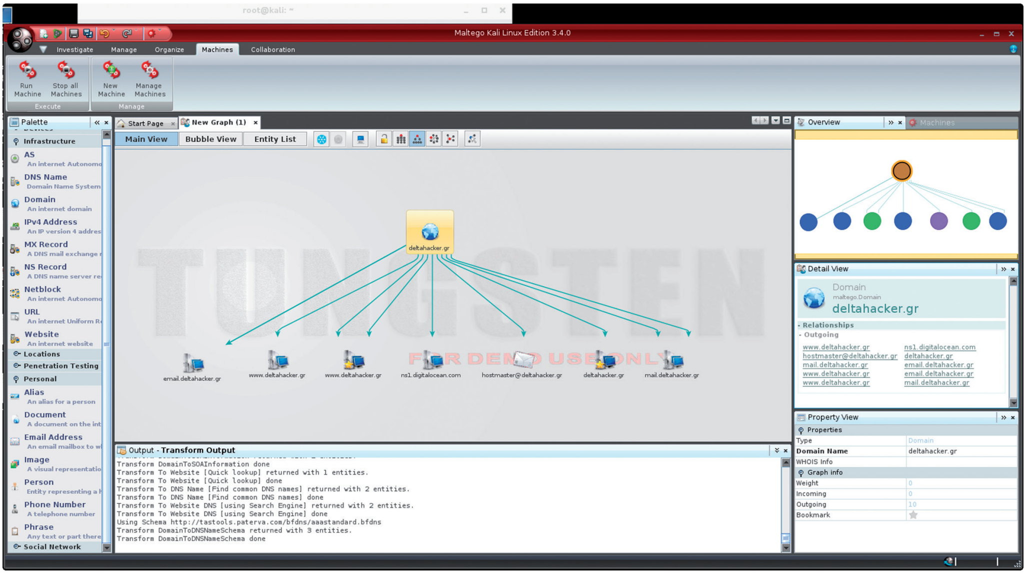
5. Στα αριστερά του παραθύρου βλέπουμε τη λεγόμενη "παλέτα", που περιλαμβάνει όλες τις διαθέσιμες επιθέσεις.



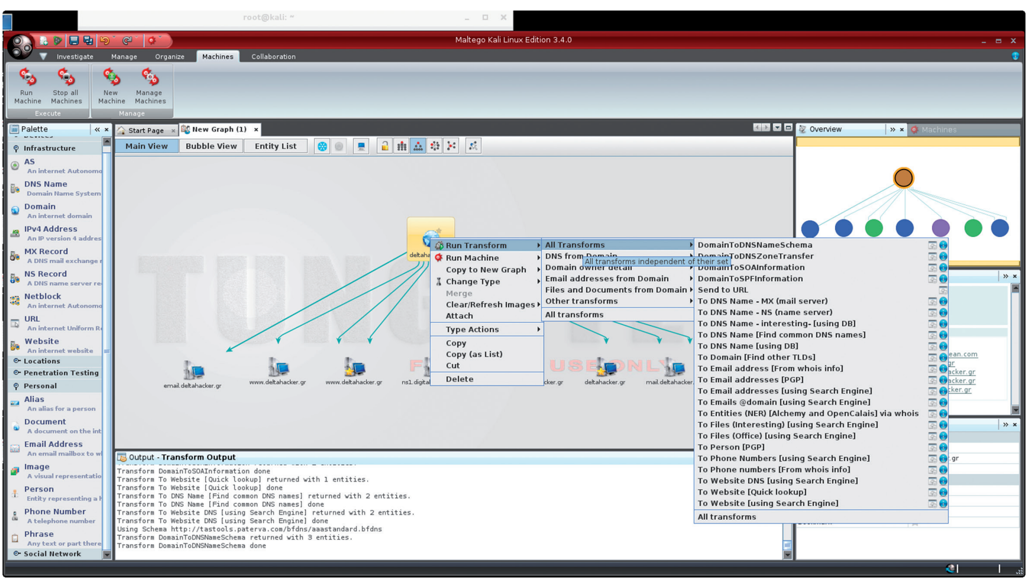
Αφού επιλέξουμε το Footprint L1 και πατήσουμε το κουμπί Next, το πρόγραμμα ζητά να δώσουμε το προς διερεύνηση domain name. Η αναζήτηση δεδομένων ξεκινά με το ένα κλικ στο Finish.

Συγκέντρωση στοιχείων από domain

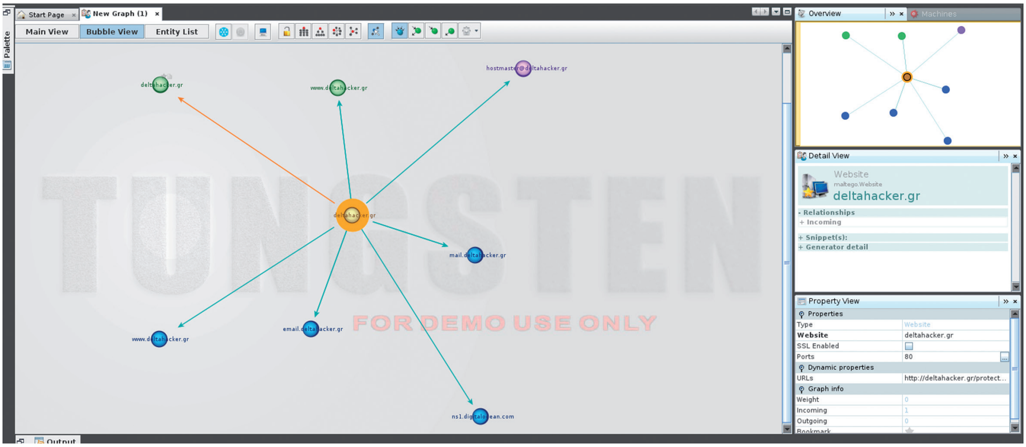
Μήπως νομίζετε ότι το μόνο που μπορεί να βρει κανείς διαθέτοντας ένα domain name είναι η αντίστοιχη διεύθυνση IP; Αν ναι, περιττό να σας πούμε ότι γελιέστε! Περίτρανη απόδειξη αποτελούν τα πλούσια ευρήματα που προσφέρει το Maltego. Για να ξεκινήσουμε μια έρευνα του είδους, μεταβαίνουμε στην καρτέλα Machines και κάνουμε κλικ στην επιλογή Run Machine. Στο παράθυρο που ανοίγει επιλέγουμε το Footprint L1. Όπως θα δείτε υπάρχουν και πρόσθετες επιλογές, που πραγματοποιούν πιο διεξοδικές έρευνες. Πάντως, για την ομαλότερη εξοικείωση με τις λειτουργίες του προγράμματος θα σας προτείναμε να ξεκινήσετε με το επίπεδο L1. Ακολούθως πατάμε Next, δίνουμε το domain name του στόχου και πατάμε Finish. Το πρόγραμμα ξεκινάει απευθείας και σε λίγα δευτερόλεπτα βλέπουμε τα πρώτα αποτελέσματα με τη μορφή ενός δενδρικού διαγράμματος. Στα αποτελέσματα της έρευνας δεν περιλαμβάνονται μόνο τα συγγενικά domain names και οι αντίστοιχες διευθύνσεις IP.



Η κεντρική σελίδα του Maltego κατά την "επίθεση" στο deltahacker.gr. Μέσα σε λίγα δευτερόλεπτα το πρόγραμμα εμφάνισε πολλά δεδομένα και όχι μόνο τη διεύθυνση IP που θα περίμενε κανείς. Μάλιστα, στην περίπτωση που είχαμε επιλέξει σαν τύπο μηχανής τα Footprint L2 ή L3, η οθόνη θα γέμιζε με εκατοντάδες πρόσθετες πληροφορίες!



Με δεξί κλικ σε κάποιον κόμβο του διαγράμματος των αποτελεσμάτων μπορούμε να ξεκινήσουμε νέες έρευνες ή και να επιστρατεύσουμε κάποιο από τα διαθέσιμα transforms. Με αυτόν τον τρόπο μπορούμε να εμπλουτίσουμε περαιτέρω τα ευρήματά μας, με μεγάλη ευκολία.



Πατώντας το κουμπί Bubble View, το διάγραμμα με τα δεδομένα αλλάζει μορφή και είναι πλέον ευκολότερο να αντιληφθούμε τις σχέσεις μεταξύ των διαφόρων ευρημάτων.

Το Maltego εντοπίζει και διευθύνσεις email που εξυπηρετούνται από το δοσμένο domain, όπως επίσης και λογαριασμούς σε κοινωνικά δίκτυα, στους οποίους το email του ιδιοκτήτη περιλαμβάνει το προς έρευνα domain! Όταν ολοκληρωθεί η βασική έρευνα μπορούμε να κάνουμε πιο εξειδικευμένες αναζητήσεις, με τη βοήθεια των transforms. Για το σκοπό αυτό αρκεί ένα δεξί κλικ σε οποιονδήποτε από τους κόμβους με το διάγραμμα των αποτελεσμάτων. Με αυτόν τον τρόπο εμφανίζονται τα διαθέσιμα κάθε φορά transforms. Εμείς κάναμε δεξί κλικ στον αρχικό κόμβο του διαγράμματος (εκείνον με το προς έρευνα domain) κι επιλέξαμε το Run Transform → Other Transforms → To Person[PGP]. Το συγκεκριμένο transform επικοινωνεί με έναν δημόσιο PGP server (keyserver) κι επιστρέφει όλα τα ονόματα και τις διευθύνσεις που σχετίζονται με το συγκεκριμένο domain. Όπως αντιλαμβάνεστε, οι επιλογές που παρέχει το πρόγραμμα είναι πάρα πολλές και με τη βοήθειά τους μπορούμε να συγκεντρώσουμε δεδομένα που θα ήταν κουραστικό και δύσκολο να συλλέξουμε χειροκίνητα.

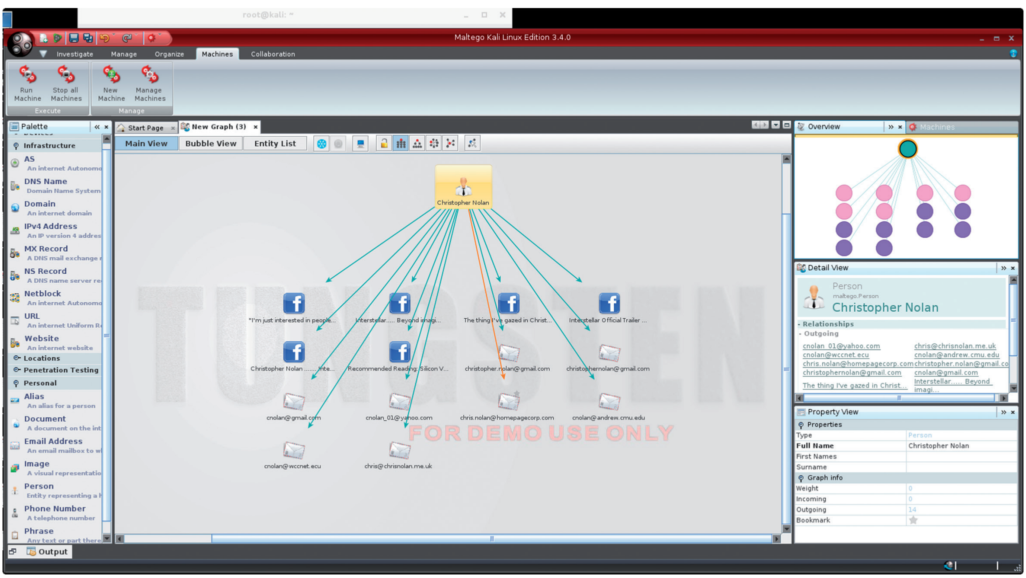
Συγκέντρωση στοιχείων με βάση το όνομα

Τι θα λέγατε αν αυτή τη φορά συγκεντρώναμε πληροφορίες γύρω από ένα όνομα; Για το σκοπό αυτό μεταβαίνουμε πάλι στην καρτέλα Machines και κάνουμε κλικ στο Run Machine. Στο παράθυρο που εμφανίζεται επιλέγουμε το "Person – Email Address", πατάμε το κουμπί Next και δίνουμε το πλήρες όνομα του ανθρώπου - στόχου. Εμείς δώσαμε το όνομα του διάσημου σεναριογράφου Christopher Nolan ;) Τα αποτελέσματα που προκύπτουν από μια έρευνα αυτού του είδους είναι πολύ περισσότερα και το Maltego σχηματίζει ένα πολύ πυκνότερο διάγραμμα. Κατά τα γνωστά, με δεξί κλικ σε οποιονδήποτε κόμβο του διαγράμματος εμφανίζεται ένα μενού με τα διαθέσιμα transforms. Έτσι, μπορούμε να επεκτείνουμε την έρευνα προς οποιαδήποτε κατεύθυνση και να αντλήσουμε δεδομένα από κοινωνικά δίκτυα ή και άλλες δικτυακές υπηρεσίες. Όπως θα διαπιστώσετε, ο όγκος των δεδομένων που συλλέγει το Maltego είναι αρκετά μεγάλος, ενώ μπορούμε να τον αυξήσουμε ακόμα περισσότερο και με μεγάλη ευκολία. Ακριβώς γι' αυτό, το πρόγραμμα παρέχει τη δυνατότητα αποθήκευσης των ευρημάτων μας για μελλοντική χρήστη. Μάλιστα,

το αρχείο που δημιουργείται είναι κρυπτογραφημένο με τον αλγόριθμο AES-128 και με ένα συνθηματικό που ορίζουμε κατά την αποθήκευση.

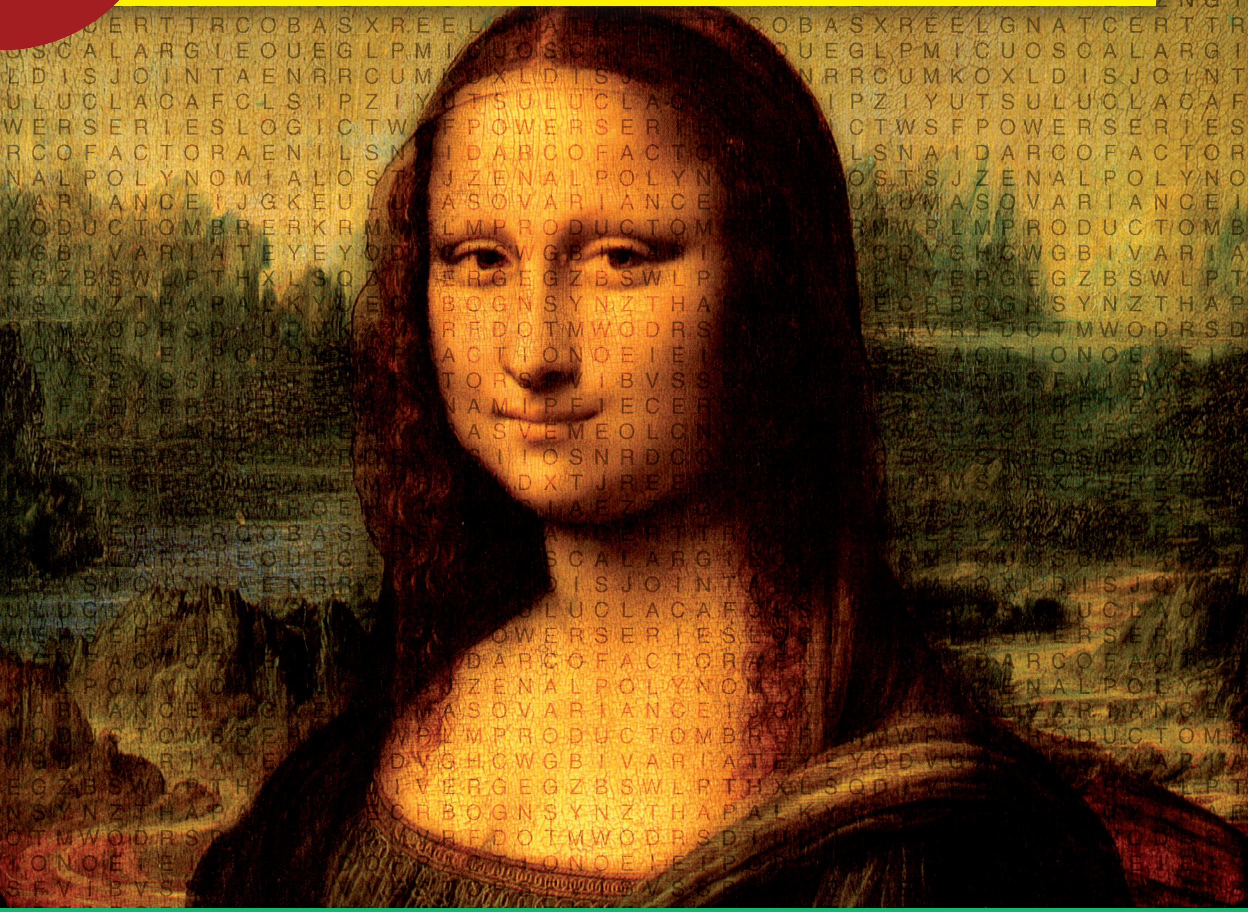
Ο πιο αδύναμος κρίκος

Αν και τα κάθε λογής συστήματα είναι ανθρώπινα δημιουργήματα κι ως εκ τούτου κληρονομούν τις ατέλειές του, η συνεχής τους εξέλιξη ενδέχεται να δυσκολέψει την εκμετάλλευση των διαφόρων ευπαθειών. Αντίθετα, ο ίδιος ο άνθρωπος είναι περισσότερο πιθανό να παραμείνει επιρρεπής σε μια καλοδουλεμένη τεχνική Social engineering. Μόνο που για να στήσουμε μια τέτοια παγίδα χρειαζόμαστε όσο το δυνατόν περισσότερα δεδομένα. Η αναζήτηση στο αχανές διαδίκτυο είναι σίγουρο ότι θα ξετρυπώσει πολλά στοιχεία που διέρρευσαν από άγνοια ή αφέλεια, αλλά θα πρέπει να γίνει σωστά και μεθοδικά. Μια απλή αναζήτηση στο Google ή σε κάποια άλλη μηχανή αναζήτησης είναι εύκολο να μας θάψει κάτω από ένα βουνό άχρηστων αποτελεσμάτων. Έτσι, η χρήση ενός εργαλείου σαν το Maltego είναι σχεδόν σωτήρια, ενώ την ίδια στιγμή θα απλοποιήσει και θα επιταχύνει την εργασία μας. Φυσικά, όπως όλα τα εργαλεία του είδους, το Maltego δεν είναι τέλειο και δεν μπορεί να κάνει τα πάντα μόνο του. Οι υπηρεσίες του είναι πολύτιμες, αλλά το πόσο θα μας βοηθήσουν εξαρτάται από το πώς θα τις χρησιμοποιήσουμε. Πιστεύουμε πάντως ότι αν ασχοληθείτε κι εσείς μαζί του, θα αρχίσετε να βλέπετε το Social engineering με άλλο μάτι.



Τα αποτελέσματα για το όνομα Christopher Nolan περιλαμβάνουν ακόμα και μερικά πιθανά emails, όπως και μερικές σελίδες που ενδέχεται να ανήκουν στο συγκεκριμένο πρόσωπο. Με δεξί κλικ επάνω στο ριζικό κόμβο του διαγράμματος (εκείνον με το όνομα) μπορούμε να ξεκινήσουμε νέες έρευνες σε περισσότερα κοινωνικά δίκτυα και να εμπλουτίσουμε τα αποτελέσματα με νέα στοιχεία.

Skill: Beginner
Tags: Steganography, Encryption, Privacy, Security, LSB



Μια εικόνα χίλιες λέξεις

Η χρήση του ρητού είναι σαφώς μεταφορική. Ή μήπως δεν είναι; Πριν απαντήσετε και πριν βιαστείτε να θεωρήσετε ότι παίζουμε με τις λέξεις, σκεφτείτε ότι μπορεί και να παίζουμε με τις εικόνες! Σας μπερδέψαμε λίγο; Όπως και να 'χει, μετά την ανάγνωση του παρόντος άρθρου οι εκφράσεις σαν αυτή του τίτλου θα αποκτήσουν νέα διάσταση.

by multiPetros

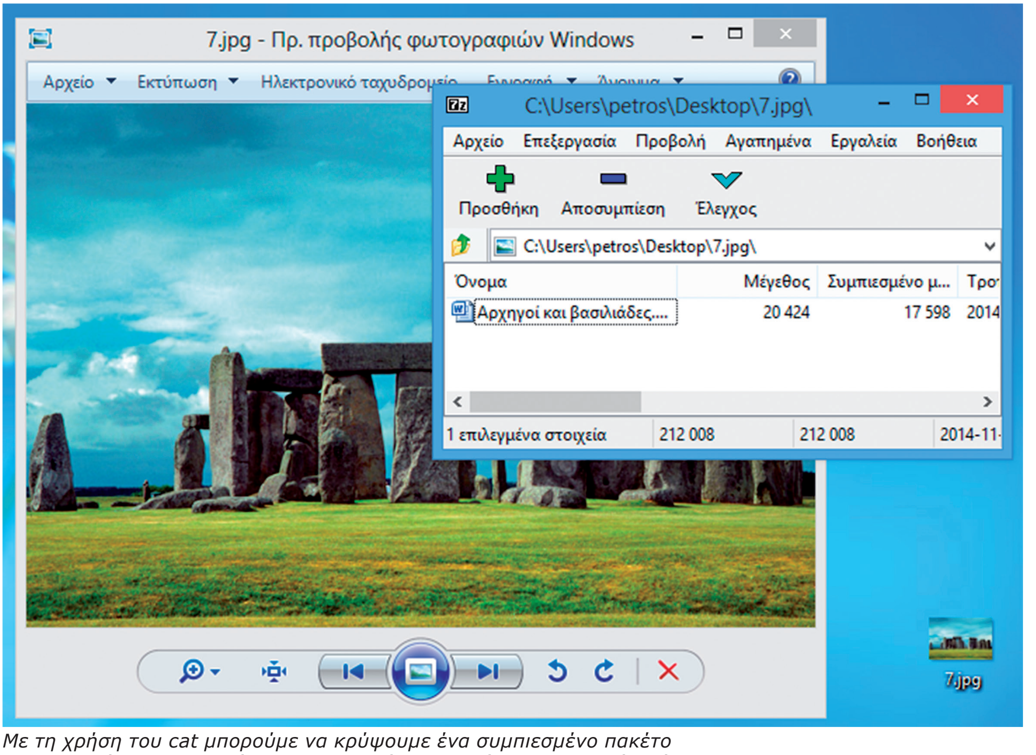
Στην εποχή της εντεινόμενης ηλεκτρονικής επιτήρησης η λεγόμενη ιδιωτικότητα έχει μπει στο στόχαστρο των απανταχού υπηρεσιών "ασφαλείας", σε σημείο μάλιστα όποιος επιδιώκει να διατηρεί ένα βασικό επίπεδο μυστικότητας των επικοινωνιών του να θεωρείται ύποπτος για τη διάπραξη εγκλημάτων. Η προπαγάνδα των διωκτικών αρχών αναπαράγεται μέσα από ευφυολογήματα του τύπου "αν δεν έχεις κάνει κάτι κακό δεν έχει τίποτα να κρύψεις" (Σ.τ.Ε. Λες και η ιδιωτικότητα δεν είναι αναφαίρετο ανθρώπινο δικαίωμα), ενώ περιλαμβάνει και δήθεν "κοινωνικά μηνύματα" που εκπαιδεύουν το κοινό στο χαφιεδισμό κι εμφανίζουν την ιδιωτικότητα σαν κάτι το επικίνδυνο. Πέρα από τα επιχειρήματα που θα μπορούσαμε να αντιτάξουμε, η ίδια η πραγματικότητα διαψεύδει τη λογική τους. Ενώ η επιτήρηση σε παγκόσμιο επίπεδο έχει αυξηθεί κατακόρυφα τα τελευταία χρόνια, οι τρομοκρατικές ενέργειες και τα εγκλήματα της μέγιστης κλίμακας (βλέπε, π.χ., πολεμικές επεμβάσεις) δεν έχουν μειωθεί καθόλου. Για την ακρίβεια έχουν αυξηθεί. Με λίγα λόγια, φαίνεται πως χτίζουμε γύρω μας μια ακόμα φυλακή, η οποία δεν προσφέρει τίποτα απέναντι σε αυτούς από τους οποίους υποτίθεται ότι προστατευόμαστε. Μήπως το πρόβλημα δεν είναι η ιδιωτικότητα, αλλά η ανικανότητα του συστήματος να υποστηρίξει μια ουσιαστική παιδεία[1] και μια ανθρωποκεντρική και δίκαιη κοινωνία; Θα σας αφήσουμε να προβληματιστείτε ελεύθερα – και ιδιωτικά, αν θέλετε :)

Στεγανογραφίας αρχή

Η κρυπτογράφηση αποτελεί την άμυνα προς κάθε επίδοξο εισβολέα της ιδιωτικής μας ζωής. Όμως τα ίδια τα κρυπτογραφημένα δεδομένα έχουν αρχίσει σιγά-σιγά να στοχοποιούνται. Μπορεί η κρυπτογράφηση να διατηρεί την ισχύ της, αλλά ταυτόχρονα λειτουργεί και σαν φάρος για την ύπαρξη σημαντικών δεδομένων. Ως αποτέλεσμα, τραβάει την προσοχή εκείνων που δεν θέλουμε να ασχολούνται μαζί μας και, προφανώς, αυτό δεν είναι και ό,τι πιο ευχάριστο. Κάπως έτσι αποφασίσαμε να ασχοληθούμε με τη στεγανογραφία. Ο συγκεκριμένος όρος προέρχεται από τις λέξεις "στεγανός" και "γραφή" και η έννοιά του θα μπορούσε να αποδοθεί ως εξής: Γραφή με συγκαλυμμένο, κρυφό τρόπο. Εν ολίγοις, ενώ η κρυπτογραφία έχει σαν αποστολή τη μετατροπή της πληροφορίας σε ακατάληπτη μορφή, η στεγανογραφία έχει ως στόχο την απόκρυψή της. Αυτό επιτυγχάνεται με το έξυπνο πακετάρισμα των δεδομένων μέσα σε κάποιο φαινομενικά αδιάφορο μέσο.

Η πρώτη αναφορά στην τεχνική της στεγανογραφίας έγινε από τον Ηρόδοτο, στην περιγραφή των περσικών πολέμων. Τα διάφορα μηνύματα που μεταφέρονταν τότε, χαράσσονταν πάνω σε ξύλινες πλάκες καλυμμένες με κερί. Σύμφωνα με τον Ηρόδοτο, λοιπόν, ο Δημάρατος αξιοποίησε τη στεγανογραφία για να ειδοποιήσει με ασφάλεια τους Σπαρτιάτες για την επικείμενη επίθεση των Περσών. Θέλοντας να αποφύγει την υποκλοπή του μηνύματος από τους Πέρσες κατασκόπους, έξυσε το κερί μιας πλάκας, έγραψε το μήνυμα πάνω στο ξύλο και στη συνέχεια επένδυσε πάλι το ξύλο με κερί. Με αυτόν τον τρόπο, η ξύλινη πλάκα έμοιαζε να μη μεταφέρει κανένα μήνυμα! Μια πιο σύγχρονη εκδοχή της τεχνικής περιλαμβάνει τη χρήση της λεγόμενης "αόρατης μελάνης". Πρόκειται για μια αόρατη ουσία Α, η οποία γίνεται ορατή όταν αντιδράσει με κάποια ουσία Β. Το μυστικό μήνυμα γραφόταν στην πίσω όψη μιας φαινομενικά αδιάφορης επιστολής, με τη χρήση της αόρατης ουσίας Α. Ο νόμιμος παραλήπτης, που γνώριζε για την παρουσία του κρυμμένου μηνύματος, άπλωνε την ουσία Β στην πίσω όψη της επιστολής και έτσι καθιστούσε το κρυφό μήνυμα ορατό!

1. Απορίες του αρχισυντάκτη: Το περιεχόμενο της παιδείας καθορίζει την κοινωνία ή η κοινωνία το περιεχόμενο της παιδείας; Ποια είναι η ρίζα των προβλημάτων τελικά; Η παιδεία ή η δομή και η λειτουργία της κοινωνίας;

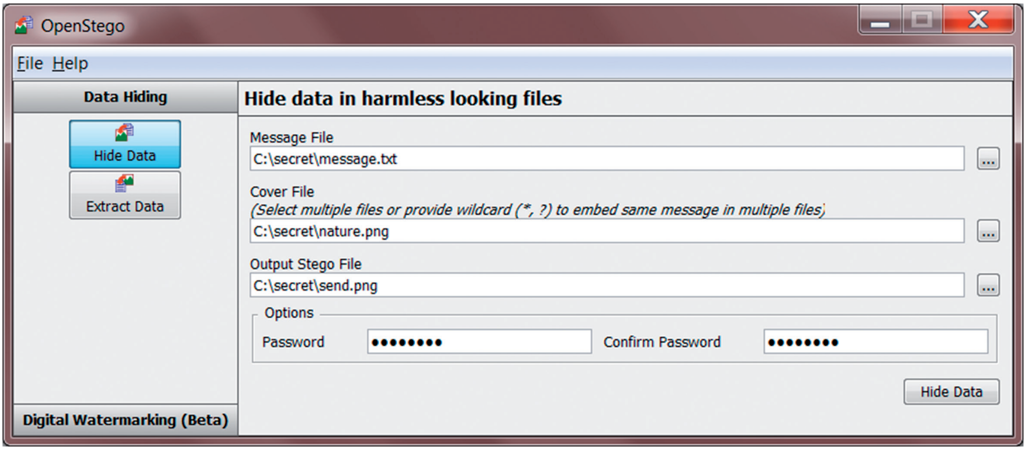


Με τη χρήση του cat μπορούμε να κρύψουμε ένα συμπιεσμένο πακέτο του 7Zip μέσα σε μια εικόνα JPEG. Η εικόνα εμφανίζεται κανονικά από τα προγράμματα προβολής και τίποτα δεν φανερώνει ότι πίσω από την όμορφη απεικόνιση του Stonehenge κρύβεται ένα συμπιεσμένο αρχείο του MS Word! Τίποτα, εκτός από το 7Zip που καταφέρνει να ανοίξει την εικόνα και να βρει ένα κείμενο του Word.

Άλλες μέθοδοι στεγανογραφίας σχετίζονται με τη χρήση προσυμφωνημένων αλγορίθμων, με τους οποίους ένα κρυφό μήνυμα κωδικοποιείται και τοποθετείται μέσα σε ένα φανερό. Πάντως, η πραγματική επανάσταση στην στεγανογραφία πραγματοποιήθηκε από τους Γερμανούς με την τεχνολογία microdots, που ανέπτυξαν κατά το Δεύτερο Παγκόσμιο Πόλεμο. Οι microdots, όπως δηλώνει και το όνομά τους, ήταν μικροσκοπικές τελείες ή σημάδια πάνω σε ασήμαντα έγγραφα, οι οποίες στην πραγματικότητα ήταν εικόνες ή και κείμενο, εκτυπωμένα σε υπερβολικά μεγάλη σμίκρυνση. Όσοι γνώριζαν την παρουσία των κρυμμένων δεδομένων μπορούσαν να τα διαβάσουν με τη βοήθεια ισχυρών μεγεθυντικών συσκευών.

Ξέρω μόνο μια γλώσσα: μηδέν και ένα

Η στεγανογραφία, όπως και η κρυπτογράφηση, απογειώνεται στον ψηφιακό κόσμο. Εκεί, φυσικοί φορείς των κρυμμένων μηνυμάτων γίνονται τα ψηφιακά δεδομένα και, γενικότερα, τα πάσης φύσεως αρχεία. Οι δε τεχνικές στεγανογράφησης περιορίζονται μόνο από τη φαντασία των προγραμματιστών, ενώ ο συνδυασμός στεγανογράφησης και κρυπτογράφησης αποτελεί ακαταμάχητο δίδυμο! Με τη στεγανογραφία μπορούμε σήμερα να μετατρέψουμε μια εικόνα, ένα αρχείο μουσικής ή ένα βίντεο σε φορέα ενός κρυμμένου και ταυτόχρονα κρυπτογραφημένου μηνύματος. Φυσικά,

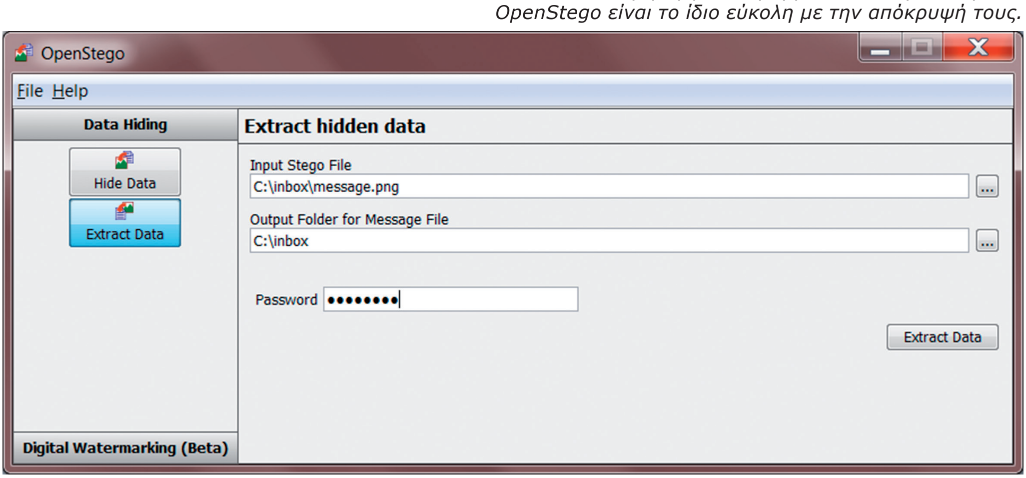


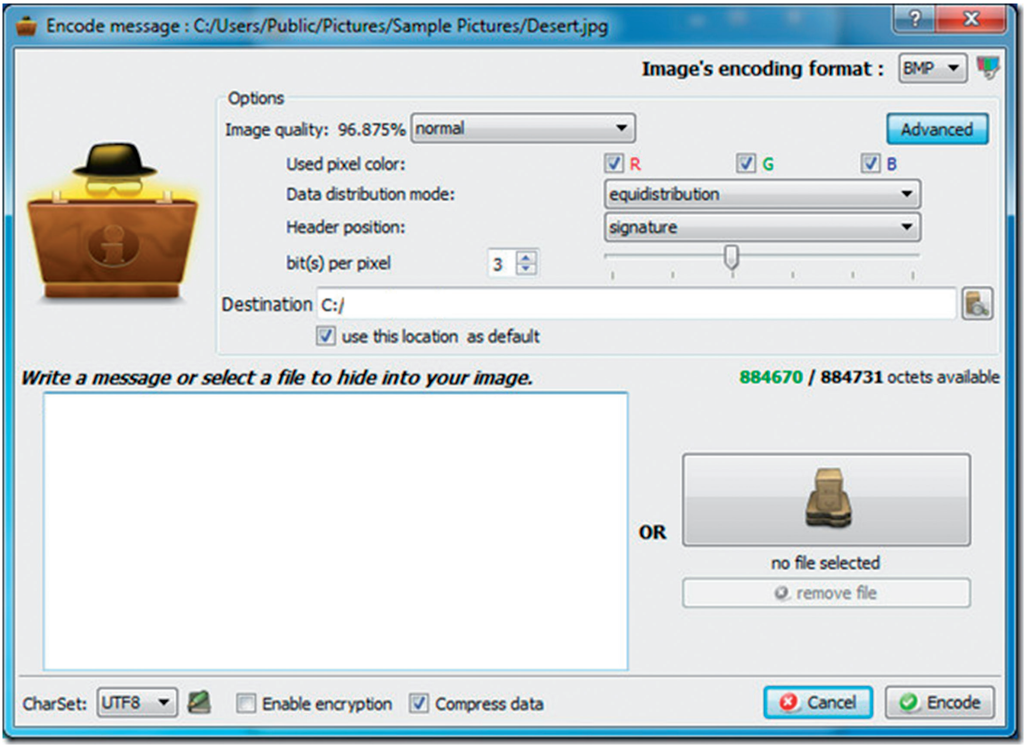
Το περιβάλλον του OpenStego είναι εξαιρετικά λιτό, όπως και οι επιλογές που παρέχει.

αυτό το μήνυμα δεν θα γίνεται αντιληπτό από αυτούς που κοιτάζουν την εικόνα ή αναπαράγουν τη μουσική και το βίντεο. Οι σύγχρονες τεχνικές στεγανογραφίας, βλέπετε, επεμβαίνουν στο δυαδικό κώδικα του αρχείου και η παρουσία των κρυφών δεδομένων δεν γίνεται αντιληπτή από τις αισθήσεις μας.

Κάποιες υλοποιήσεις της στεγανογραφίας εκμεταλλεύονται το χώρο που περιλαμβάνουν τα πολυμεσικά αρχεία για τα λεγόμενα metadata. Είναι σίγουρο ότι έχετε δει πολλά προγράμματα να παρουσιάζουν πληροφορίες για τον καλλιτέχνη και το όνομα του δίσκου από τον οποίο προέρχεται ένα μουσικό κομμάτι. Σε άλλες περιπτώσεις βλέπουμε πληροφορίες για την ταχύτητα του κλείστρου, το μοντέλο της φωτογραφικής μηχανής, την ευαισθησία του αισθητήρα στο φως κατά τη λήψη μιας φωτογραφίας κ.ο.κ. Οι πληροφορίες αυτού του είδους (ID3 tags στη μία περίπτωση και EXIF data στην άλλη) αντλούνται από τα ίδια τα αρχεία. Ε, λοιπόν, ορισμένες τεχνικές στεγανογραφίας τοποθετούν τα μηνύματά μας σε κρυπτογραφημένη μορφή, στο χώρο που προβλέπεται για τα metadata. Όπως καταλαβαίνετε, αυτό το

Η εξαγωγή των κρυμμένων δεδομένων με το OpenStego είναι το ίδιο εύκολη με την απόκρυψή τους.



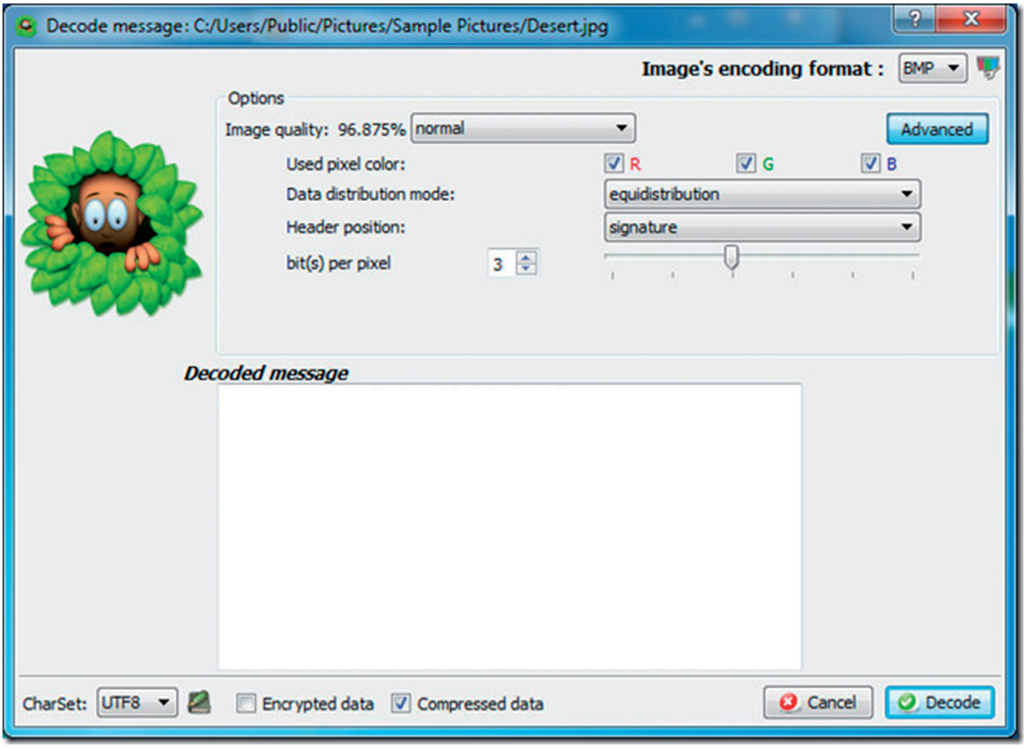


Οι ρυθμίσεις που προσφέρει το SilentEye είναι πάμπολλες και η εφαρμογή επιτρέπει στο χρήστη να παρέμβει σε χαμηλό επίπεδο.

τέχνασμα είναι ιδιαίτερα απλοϊκό κι έχει πολλά ελαττώματα. Τα πεδία των αρχείων που φιλοξενούν τα metadata έχουν σταθερό και σχετικά μικρό μήκος, γεγονός που επιβάλλει αυτόματα έναν ασφυκτικό περιορισμό για το μέγεθος των κρυφών μηνυμάτων μας. Εξάλλου, καθώς τα metadata προβάλλονται από τα περισσότερα προγράμματα αναπαραγωγής, αν αλλοιώσουμε τα περιεχόμενά τους θα κινήσουμε αμέσως τις υποψίες των χρηστών. Κάποιες βελτιωμένες τεχνικές στεγανογραφίας αξιοποιούν την εσωτερική δομή των πολυμεσικών αρχείων, επεμβαίνοντας σε "χαμηλότερο επίπεδο". Ουσιαστικά, αυτές οι τεχνικές εκμεταλλεύονται τη δομή που επιβάλλει το εκάστοτε container format (AVI, MP3, MP4, MKV κ.ά.) και παρεμβάλλουν τα δεδομένα μας με τη μορφή ολόκληρων μπλοκ, τα οποία ωστόσο παρακάμπτονται κατά την αναπαραγωγή και η παρουσία τους δεν γίνεται αντιληπτή.

Εικόνες, χρώματα και κρυφά καμώματα

Το χρώμα κάθε pixel στις εικόνες αναπαρίσταται αριθμητικά από ένα πλήθος bit. Όσο πιο πολλά τα bits που χρησιμοποιούνται για την περιγραφή του χρώματος, τόσο πιο πολλά είναι και τα χρώματα και, όπως συνηθίζουμε να λέμε, τόσο πιο μεγάλο είναι το βάθος χρώματος. Σήμερα, για την περιγραφή του χρώματος ενός pixel χρησιμοποιούνται συνήθως 24 bits. Κι επειδή για την περιγραφή του χρώματος χρησιμοποιούνται συνήθως τρία βασικά χρώματα (η γνωστή τριάδα RGB), σε καθένα από αυτά τα χρώματα αντιστοιχούν 8 bits. Για παράδειγμα, η δυαδική αναπαράσταση ενός κατακόκκινου pixel έχει ως εξής:



Το παράθυρο για την ανάκτηση των στεγανογραφημένων δεδομένων του SilentEye έχει εξίσου πολλές ρυθμίσεις με εκείνο της κωδικοποίησης.

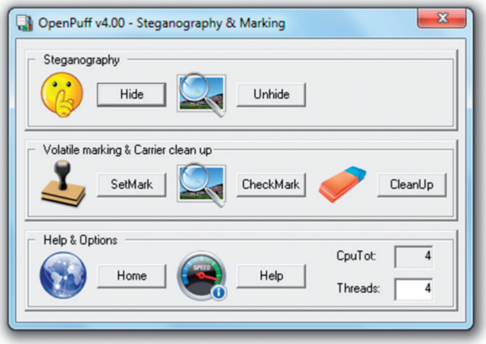
```
11111111 00000000 00000000
```

Η πρώτη οκτάδα περιγράφει το κόκκινο (red) και έχει τη μέγιστη τιμή, ενώ οι επόμενες δύο περιγράφουν το πράσινο και το μπλε αντίστοιχα. Όπως καταλαβαίνετε, οι "γειτονικές" αποχρώσεις παρουσιάζουν μικρή διαφορά μεταξύ τους, που συνήθως είναι ανεπαίσθητη για το ανθρώπινο μάτι. Αν μεταβάλλαμε ελαφρώς την παραπάνω απόχρωση, το pixel θα φαινόταν στα μάτια μας εξίσου κόκκινο. Αυτή η αλλοίωση στην απόχρωση θα μπορούσε να γίνει τροποποιώντας το λιγότερο σημαντικό bit (LSB) σε κάθε οκτάδα. Έτσι, η παραπάνω τιμή θα έπαιρνε την ακόλουθη μορφή:

```
11111110 00000001 00000001
```

Τι μπορεί να συμπεράνει κανείς από εδώ; Εφόσον οι δύο αποχρώσεις φαίνονται πανομοιότυπες στο ανθρώπινο μάτι, θα μπορούσαμε να αλλοιώσουμε όλα τα LSB όλων των οκτάδων, για όλα τα pixel της εικόνας, χωρίς να γίνει ορατή η επέμβασή μας. Με άλλα λόγια, θα μπορούσαμε ενσωματώσουμε στην εικόνα μας 3bit πληροφορίας για κάθε pixel!

Σ' αυτή την ιδέα βασίζονται αρκετά προγράμματα στεγανογραφίας, τα οποία καταφέρνουν να ενσωματώνουν ολόκληρα κείμενα μέσα σε εικόνες. Η προσθήκη των νέων δεδομένων εισάγει στην εικόνα ελάχιστο οπτικό θόρυβο, ο οποίος περνάει απαρατήρητος από το ανθρώπινο μάτι. Φυσικά, τα στεγανογραφημένα δεδομένα αποθηκεύονται συνήθως σε κρυπτογραφημένη μορφή κι αυτό καθιστά την "εξόρυξή" τους πολύ δύσκολη. Εξάλλου, ορισμένοι αλγόριθμοι στεγανογραφίας εισάγουν ένα



Στο λιτό αρχικό παράθυρο του προγράμματος, τίποτα δεν φανερώνει τη δύναμη και την προστασία που μπορεί να προσφέρει το OpenPuff!

πρόσθετο "ανακάτωμα" στη σειρά με την οποία αποθηκεύονται τα δεδομένα, ώστε να θολώσουν τα νερά ακόμα περισσότερο.

Πίσω από τις νότες, τα δικά μας bits κόβουν βόλτες

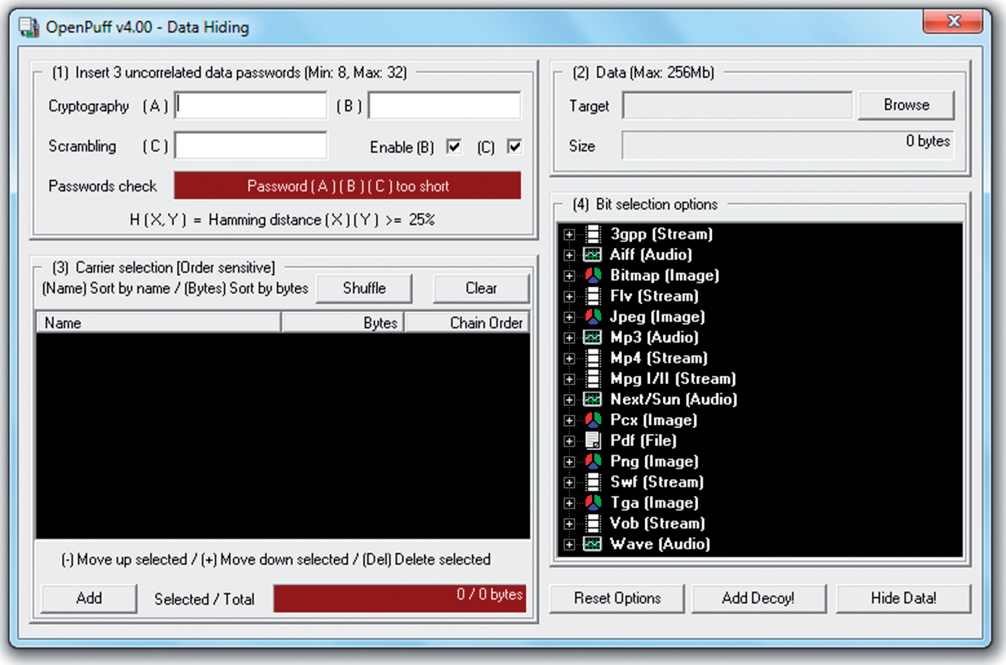
Στα αρχεία ήχου δεν έχουμε pixel, ούτε χρώματα. Έχουμε όμως ψηφιοποιημένα δείγματα του ήχου, καθένα από τα οποία αναπαρίσταται από μια αριθμητική τιμή. Το πλήθος αυτών των δειγμάτων είναι συνήθως αυξημένο και κατ' επέκταση τα αντίστοιχα αρχεία ήχου είναι μεγάλα. Γι' αυτό το λόγο εφαρμόζονται διάφορες τεχνικές που είτε περιορίζουν το πλήθος των δειγμάτων (απόρριψη πληροφορίας), είτε την έκταση που καταλαμβάνουν (συμπίεση). Το ανθρώπινο αφτί, βλέπετε, έχει πολλές ατέλειες. Ακούει ένα περιορισμένο εύρος συχνοτήτων, ενώ η ευαισθησία του σε μια οποιαδήποτε συχνότητα καθορίζεται από τις υπόλοιπες συχνότητες που ακούγονται την ίδια στιγμή. Ορισμένοι αλγόριθμοι εκμεταλλεύονται αυτές τις ιδιαιτερότητες της ακοής και απομακρύνουν ένα μεγάλο πλήθος δειγμάτων, χωρίς να παραμορφώσουν αισθητά τον ήχο. Αναφερόμαστε στους λεγόμενους απωλεστικούς αλγορίθμους συμπίεσης (βλέπε MP3, AAC κ.λπ.). Αναρωτιέστε γιατί μιλάμε ξαφνικά για τον ήχο και τη συμπίεσή του; Οι τεχνικές στεγανογραφίας για την τοποθέτηση μηνυμάτων σε αρχεία μουσικής στηρίζονται στις ίδιες αρχές με τους απωλεστικούς αλγόριθμους: απορρίπτουν ένα μέρος των ηχητικών δεδομένων και στη θέση τους τοποθετούν τα τμήματα του κρυφού μηνύματος. Βέβαια, αυτή η τεχνική δεν μπορεί να εφαρμοστεί σε αρχεία ήχου που έχουν υποστεί ήδη κάποια απωλεστική συμπίεση. Υπάρχουν όμως και πιο εξεζητημένες μέθοδοι, που επιτρέπουν την ενσωμάτωση κρυφών μηνυμάτων ακόμα και σε MP3.

Στεγανογραφία εδώ και τώρα!

Μια ταχύτατη αλλά αδύναμη μορφή στεγανογραφίας μπορούμε να πετύχουμε από την κονσόλα οποιουδήποτε συστήματος Unix/Linux, με τη βοήθεια των εργαλείων cat και unzip. Με τη βοήθειά τους μπορούμε να κρύψουμε αρχεία ZIP μέσα σε αρχεία εικόνας PNG, JPEG αλλά και GIF ("σταθερά" ή animated). Πρώτα απ' όλα χρειάζεται να δημιουργήσουμε ένα αρχείο ZIP με τα δεδομένα που θέλουμε να κρύψουμε. Όπως αντιλαμβάνεστε, πριν από τη συμπίεση θα ήταν καλό να πραγματοποιήσουμε και κάποιο είδος κρυπτογράφησης, ώστε τα δεδομένα μας να είναι όσο το δυνατόν πιο ασφαλή, ακόμα και στην περίπτωση που η παρουσία τους αποκαλυφθεί. Στη συνέχεια πρέπει να επιλέξουμε την εικόνα που θα λειτουργήσει σαν φορέας των κρυφών δεδομένων μας. Εδώ ενδείκνυται η επιλογή μιας εικόνας με μεγάλη ανάλυση, ώστε το αυξημένο της μέγεθος να μην κινεί υποψίες. Αν υποθέσουμε ότι το πακέτο με το κρυπτογραφημένο μήνυμα ονομάζεται data.zip και η εικόνα image.zip, αρκεί να δώσουμε κάτι τέτοιο:

```
cat image.jpg data.zip > carrier_image.jpg
```

Σχεδόν αμέσως θα πάρουμε την εικόνα carrirer_image.jpg, που θα προβάλλεται



Πολλές μορφές αρχείων – φορέων και ένα παρανοϊκό επίπεδο ασφάλειας, για έναν (εξίσου) παρανοϊκό κόσμο :D

χωρίς το παραμικρό πρόβλημα από οποιονδήποτε image viewer ή και web browser. Μέσα σε αυτό το αρχείο, όμως, θα έχει ενσωματωθεί και ολόκληρο το πακέτο ZIP. Για να το εξάγουμε πρέπει να στραφούμε πάλι στη γραμμή εντολών και να δώσουμε:

```
unzip carrier_image.jpg
```

Τόσο απλά! Το πρόγραμμα unzip ενδέχεται να διαμαρτυρηθεί για την παρουσία παραπανίσιων και άχρηστων δεδομένων, αλλά θα εντοπίσει και θα αποσυμπιέσει κανονικά τα δεδομένα του data.zip. Φυσικά, μπορούμε να παρακάμψουμε τη χρήση της κονσόλας και να κάνουμε την εξαγωγή με τον αγαπημένο μας archive manager. Επομένως, η εξαγωγή του κρυφού "πακέτου" θα μπορούσε να πραγματοποιηθεί και από τα Windows, με τη βοήθεια του αγαπημένου μας 7Zip. Παρεμπιπτόντως, μιας και αναφέραμε το παντοδύναμο 7Zip, αξίζει να σημειώσουμε ότι το κολπάκι μας λειτουργεί εξίσου καλά και με συμπιεσμένα αρχεία 7z. Πάντως, η απόκρυψη ενός αρχείου μέσα σε κάποιο άλλο με τη βοήθεια του cat δεν συνιστά ασφαλή λύση και θα λειτουργήσει μόνο στα μάτια των πιο άπειρων χρηστών.

Κρυφτούλι για όλους

Αφήνοντας πίσω τους ερασιτεχνισμούς, μπορούμε να περάσουμε σε μερικά "σοβαρά" εργαλεία στεγανογραφίας. Ένα από αυτά είναι το OpenStego, που διατίθεται για Windows και για Linux κι όπως δηλώνει το όνομά του, αποτελεί προϊόν ανοιχτού κώδικα. Μπορούμε να το κατεβάσουμε από τον επίσημο ιστότοπο στη διεύθυνση www.openstego.info. Το OpenStego είναι ιδιαίτερα εύχρηστο και στο περιβάλλον του δεσπόζουν δύο κουμπιά για τις δύο κύριες εργασίες: Την ενσωμάτωση και απόκρυψη

αρχείων μέσα σε άλλα, καθώς και την ανάκτηση κρυμμένων δεδομένων. Το εν λόγω εργαλείο πραγματοποιεί και κρυπτογράφηση με τον αλγόριθμο DES, με βάση ένα password που ορίζει κάθε φορά ο χρήστης. Δυστυχώς, το OpenStego μπορεί να κρύψει αρχεία μόνο μέσα σε εικόνες.

Μια άλλη εφαρμογή στεγανογραφίας που κέντρισε το ενδιαφέρον μας είναι το (www.silenteye.org). Πρόκειται για ένα ελαφρώς ικανότερο πρόγραμμα σε σχέση με το OpenStego, που διατίθεται σε πακέτα για Windows, OS X και DEB για το Linux. Το SilentEye υποστηρίζει αρχεία εικόνας BMP και JPEG, καθώς και αρχεία ήχου της μορφής WAVE/PCM. Όπως και το προηγούμενο εργαλείο, το SilentEye διαθέτει εξαιρετικά λιτό περιβάλλον και περιλαμβάνει μόνο δύο κουμπιά για τις δύο κύριες λειτουργίες. Ωστόσο, το συγκεκριμένο πρόγραμμα προσφέρει στο χρήστη και πολλές επιλογές. Ανάμεσά τους ξεχωρίσαμε τη δυνατότητα ρύθμισης της ποιότητας του τελικού αρχείου, τη δυνατότητα παρεμβολής δεδομένων σε συγκεκριμένα μόνο χρώματα του μοντέλου RGB και τη δυνατότητα επιλογής του πλήθους των bits που θα τροποποιούνται, ανά χρώμα κι ανά pixel. Αυτές οι ρυθμίσεις μοιάζουν αδιάφορες, αλλά διαφοροποιούν αρκετά τον αλγόριθμο στεγανογράφησης και δυσκολεύουν τον εντοπισμό των κρυφών δεδομένων.

Ένα τρίτο εργαλείο που κέρδισε την προσοχή μας είναι το πανίσχυρο OpenPuff (http://embeddedsw.net/OpenPuff_Steganography_Home.html), από τον Cosimo Oliboni. Αν και το όνομά του παραπέμπει σε κάποιο έργο ανοιχτού κώδικα, πρόκειται για πρόγραμμα με "υβριδική" φύση: Το τμήμα που αναλαμβάνει την κρυπτογράφηση (libObfuscate) είναι ανοιχτού κώδικα, ενώ το υπόλοιπο δεν είναι! Πάντως η εφαρμογή διατίθεται δωρεάν και η αναδιανομή της επιτρέπεται. Σε αντίθεση με τα άλλα δύο εργαλεία, το OpenPuff υποστηρίζει μια ευρύτατη γκάμα αρχείων που μπορούν να λειτουργήσουν ως φορείς για τα δεδομένα μας:

- Αρχεία εικόνας (bmp, jpg, png και tga)
- Αρχεία ήχου (wav, mp3 και aiff)
- Αρχεία βίντεο (mpeg, mpeg2, 3gp, mp4 και vob)
- Formats της Adobe (pdf, swf και flv)

Μάλιστα, σε ό,τι αφορά την κρυπτογράφηση, θα μπορούσαμε να πούμε ότι το εν λόγω εργαλείο συμπεριφέρεται παρανοϊκά. Ούτε λίγο ούτε πολύ, το OpenPuff επιτρέπει την πολλαπλή κρυπτογράφηση με χρήση διαφορετικών μυστικών κλειδιών για κάθε "επίπεδο" κρυπτογράφησης, ενώ επιβάλλει πάντα και ένα πρόσθετο "ανακάτωμα" στη σειρά αποθήκευσης των δεδομένων.

Ψήλοι στ' αυτιά μου μπήκανε

Μετά από τη μικρή μας περιήγηση στα κατατόπια της στεγανογραφίας, είναι σίγουρο ότι κανένας δεν θα βλέπει με το ίδιο μάτι τα (φαινομενικά) αδιάφορα αρχεία εικόνας και ήχου που συναντάμε στο Internet. Εμείς τουλάχιστον επηρεαστήκαμε τόσο βαθιά, που αρχίσαμε να εξετάζουμε μία προς μία τις εικόνες στο site του Parabing ;)

Skill: Intermediate
Tags: Network scanning, port scanning, brute force, session hijacking, MiTM

ZIMPERIUM

by Cr0wTom

Επαγγελματικό, δωρεάν και πανίσχυρο!

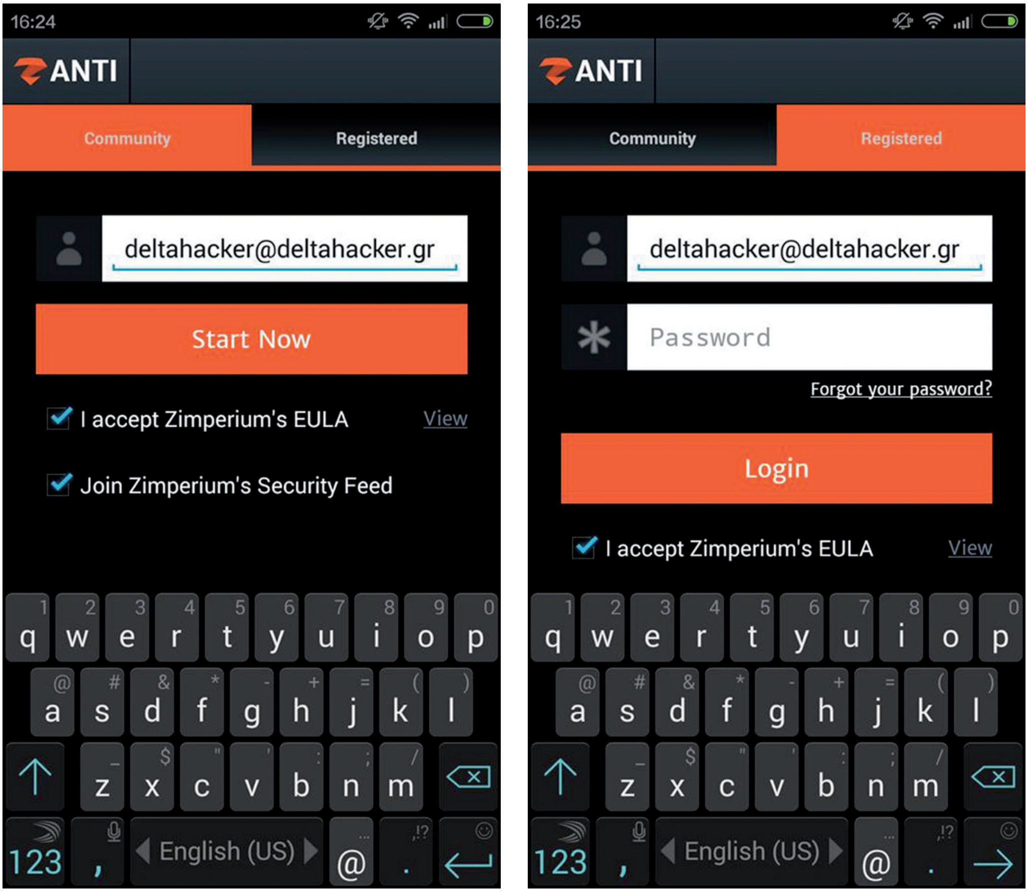
Οι εφαρμογές του Android που σχετίζονται με το penetration testing δεν είναι λίγες. Μέχρι πρότινος, όμως, καμία από τις δωρεάν λύσεις δεν μπορούσε να συναγωνιστεί τις εμπορικές. Αυτό το κενό ήρθε να καλύψει η Zimperium με μια νέα σειρά εφαρμογών και υπηρεσιών. Συνεχίστε να διαβάζετε και κάτι μάς λέει ότι σύντομα θα αποκτήσετε κι εσείς το δικό μας ενθουσιασμό :D

Στο άρθρο μας με τίτλο "Hacking with Android", στο τεύχος 035, πραγματοποιήσαμε μια εκτενή αναφορά στα εργαλεία που διευκολύνουν ή αυτοματοποιούν το penetration testing από την πλατφόρμα του Android. Για την προετοιμασία εκείνου του άρθρου είχαμε, μεταξύ άλλων, σαρώσει το Web, σε μια προσπάθεια να βρούμε, να γνωρίσουμε και τελικά να παρουσιάσουμε τις εφαρμογές που θεωρούσαμε καλύτερες. Δεν σας κρύβουμε μάλιστα ότι μετά από εκείνο το άρθρο είχαμε την αφέλεια να πιστεύουμε ότι το ζήτημα εξαντλήθηκε. Να όμως που μια εταιρεία αποφάσισε να μας διαψεύσει! (Σ.τ.Ε. Δηλαδή δεν είχε τι καλύτερο να κάνει;! Εντυπωσιασμένος.) Ο λόγος γίνεται για την Zimperium, που ενδέχεται να τη θυμόσαστε ως τη δημιουργό του εργαλείου zAnti. Πρόκειται για μια συνδρομητική σουίτα με εργαλεία για penetration testing, που δυστυχώς δεν κατάφερε ποτέ να μας ενθουσιάσει. Ε, λοιπόν, τον τελευταίο καιρό η εταιρεία δημιούργησε ένα σύνολο εφαρμογών και υπηρεσιών που όχι μόνο τράβηξαν την προσοχή μας, αλλά μάς εντυπωσίασαν κιόλας.

Πριν από λίγες εβδομάδες, ο δημιουργός του αγαπημένου μας dSploit, ο Simone Margaritelli, ανακοίνωσε τη συγχώνευση του πνευματικού του τέκνου με το εργαλείο zAnti, που είχε σαν αποτέλεσμα τη δημιουργία του zAnti 2. Πριν προχωρήσουμε στην παρουσίασή μας, αξίζει να πούμε δυο λόγια για τη Zimperium. Η εταιρία ιδρύθηκε τον Οκτώβριο του 2010 και ένα χρόνο μετά παρουσίασε το πρώτο της εργαλείο για το Android, με το όνομα ANTi (Android Network Toolkit). Πολύ γρήγορα, το πρόγραμμα μετονομάστηκε σε zAnti και αποτελούσε μια επαγγελματική λύση για επιχειρήσεις που ήθελαν να τσεκάρουν την ασφάλεια των δικτύων τους. Μάλιστα, για εκείνη την εποχή ήταν ένα αξιόλογο εργαλείο. Η τεχνολογία όμως προχωρούσε και το zAnti έμεινε στάσιμο για αρκετά χρόνια. Αυτά έως λίγες εβδομάδες πριν, όπου εμφανίστηκε η ανανεωμένη έκδοση zAnti 2 και στο οπλοστάσιο της εταιρείας προστέθηκαν δύο νέες εφαρμογές: Το zIPS και το zConsole. Το πρώτο αποτελεί ένα εργαλείο εντοπισμού και απόκρουσης επιθέσεων, ενώ το δεύτερο λειτουργεί σαν μια εφαρμογή καταγραφής των συμβάντων που σχετίζονται με την ασφάλεια μιας συσκευής. Αλλά κάπου εδώ πρέπει να αφήσουμε τον πρόλογο και να περάσουμε σε μια αναλυτική παρουσίαση των δυνατοτήτων που προσφέρει κάθε εφαρμογή.

Ένα πολύπλευρα αναβαθμισμένο dSploit

Το zAnti 2 δεν έχει καμία σχέση με τον πρόγονό του. Ακολουθεί τα χνάρια του πολύ πετυχημένου dSploit, ενώ ταυτόχρονα διακρίνεται για την άκρως επαγγελματική και προσεγμένη του υλοποίηση. Συνήθως, όταν ακούμε για ισχυρές επαγγελματικές λύσεις, το μυαλό μας τρέχει σε αδικαιολόγητα υψηλές χρεώσεις, που απευθύνονται σε μεγάλες εταιρείες και σίγουρα όχι στη δική μας τσέπη. Κάτι τέτοιο ίσχυε για το zAnti, αλλά όχι και για το zAnti 2. Η ανανεωμένη εφαρμογή διατίθεται δωρεάν και μπορούμε να την κατεβάσουμε από εδώ: www.zimperium.com/zanti-mobile-penetration-testing. Φυσικά, η Zimperium πουλάει και άδειες χρήσης που προορίζονται για μεγάλους πελάτες (εταιρείες), αλλά η δωρεάν εκδοχή του προγράμματος δεν θέτει κανέναν περιορισμό – ή τουλάχιστον κανέναν *ενοχλητικό* περιορισμό. Όσοι έχουν αγοράσει λογαριασμό μπορούν να εκμεταλλευτούν τις διάφορες ευπάθειες που εντοπίζουν με τη βοήθεια των server της εταιρείας. Οι υπόλοιποι μπορούν να πραγματοποιήσουμε οποιαδήποτε άλλη ενέργεια, εκτός από το να επιστρατεύσουν τα μηχανήματα της Zimperium. Όπως αντιλαμβάνεστε αυτός ο περιορισμός δεν μας ενοχλεί καθόλου, αφού κατά τα άλλα η εφαρμογή λειτουργεί απρόσκοπτα και παρέχει ένα πλούσιο σύνολο δυνατοτήτων. Θα λέγαμε ότι το zAnti 2 περιλαμβάνει τις περισσότερες από τις λειτουργίες του dSploit και κατά κάποιο τρόπο τις αυτοματοποιεί. Μη νομίζετε όμως ότι σταματάει εκεί…

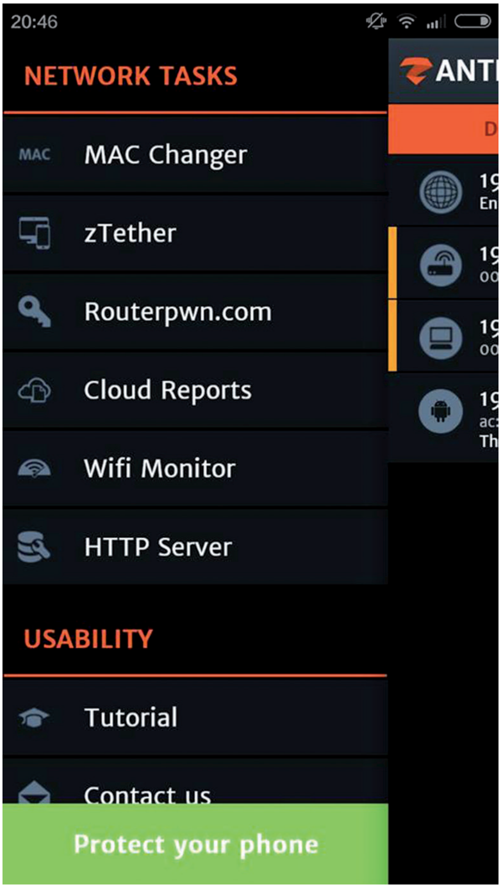


Ξεκινώντας το zAnti 2 πρέπει να δηλώσουμε ένα λογαριασμό email για να συνδεθούμε στην εφαρμογή δωρεάν. Εναλλακτικά, θα μπορούσαμε να αγοράσουμε μια από εκείνες τις πανάκριβες συνδρομές στην εταιρεία και να δώσουμε τα στοιχεία του σχετικού λογαριασμού...

Όταν εκτελούμε την εφαρμογή εμφανίζονται οι καρτέλες Community και Registered. Για όσους δεν διαθέτουν λογαριασμό, η εκκίνηση της εφαρμογής πραγματοποιείται από την καρτέλα Community. Στην κεντρική "σελίδα" της εφαρμογής βρίσκουμε το network scanner, που ενεργοποιείται αυτόματα και σκανάρει το ασύρματο δίκτυο στο οποίο είμαστε συνδεδεμένοι. Όπως αναφέραμε ήδη, το zAnti 2 αυτοματοποιεί αρκετές εργασίες σε σχέση με τους προγόνους του. Έτσι, όταν ολοκληρώνεται το σκανάρισμα του δικτύου δεν εμφανίζεται μόνο μια λίστα με τις συσκευές. Για καθεμία εμφανίζονται οι διευθύνσεις IP και MAC, οι ανοικτές δικτυακές θύρες και το λειτουργικό σύστημα. Στην πρώτη θέση της λίστας συναντάμε πάντα το subnet mask του δικτύου, ενώ στη δεύτερη εμφανίζεται πάντοτε ο router του δικτύου (network getaway). Στη μπάρα πάνω από τη λίστα με τις συσκευές συναντάμε τρείς επιλογές: Το ιστορικό των συσκευών που έχουν ανιχνευθεί κατά περιόδους, την επανεκκίνηση του network scanner και τη χειροκίνητη προσθήκη κάποιου host. Στην ίδια μπάρα εμφανίζεται και το εικονίδιο του zAnti, που κρύβει πίσω του ένα ολόκληρο μενού. Το μενού είναι χωρισμένο στα τμήματα Network Tasks και Usability. Το δεύτερο τμήμα

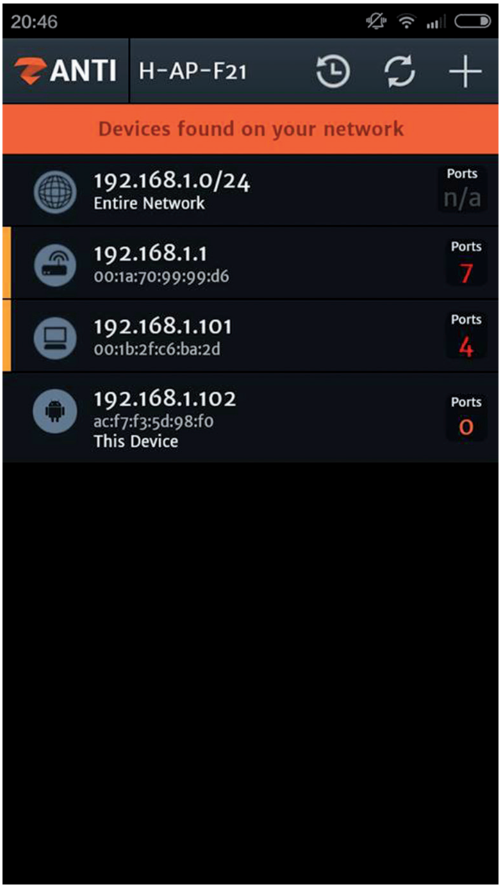
περιλαμβάνει μια παραπομπή προς τις γενικές ρυθμίσεις του προγράμματος, καθώς και διάφορους τρόπους επικοινωνίας με την εταιρεία. Στο πρώτο τμήμα, όμως, εμφανίζονται μερικές άκρως ενδιαφέρουσες επιλογές. Ας δούμε μερικές από αυτές:

- **MAC Changer:** Προφανώς, από εδώ μπορούμε να αλλάξουμε την διεύθυνση MAC της συσκευής μας. Κάτι τέτοιο μπορεί να μας φανεί χρήσιμο είτε για λόγους που σχετίζονται με την ανωνυμία, είτε για τη διεξαγωγή κάποιας επίθεσης.
- **zTether:** Αυτή η επιλογή δημιουργεί ένα rogue access point. Με απλά λόγια, δημιουργεί ένα ασύρματο δίκτυο που μπορεί να εκτελέσει επιθέσεις Man-in-The-Middle, σε οποιονδήποτε (κάνει το λάθος να) συνδεθεί.
- **Routerpwn.com:** Αυτό δεν αποτελεί κάποια ενσωματωμένη λειτουργία του προγράμματος, αλλά μια συντόμευση προς τη σχετική υπηρεσία. Στο routerpawn έχουμε αναφερθεί και παλιότερα. Να θυμίσουμε ότι πρόκειται για μια δικτυακή εφαρμογή, που περιέχει όλες τις γνωστές ευπάθειες των router και τα αντίστοιχα exploits!
- **WiFi Monitor:** Όπως καταλαβαίνετε, από εδώ μπορούμε να αντλήσουμε πληροφορίες για τα γειτονικά ασύρματα δίκτυα. Για παράδειγμα, μπορούμε να μάθουμε τις MAC address των access points, την ισχύ του σήματος καθενός, τα κανάλια στα οποίο εκπέμπουν άλλα δίκτυα, τα SSID και τα πρωτόκολλα ασφαλείας.



Το πιο ενδιαφέρον τμήμα του μενού του zAnti 2. Πρόκειται για τη λίστα με τα διαθέσιμα δικτυακά εργαλεία, η οποία θα φανεί χρήσιμη σε οποιονδήποτε ασχολείται με το Penetration Testing. Το μενού συνεχίζει με ορισμένες επιλογές για την επικοινωνία μας με την εταιρεία καθώς και τις γενικές ρυθμίσεις της εφαρμογής.

Αν επιλέξουμε μια συσκευή από την κεντρική "σελίδα" του προγράμματος, μεταφερόμαστε σε μια νέα περιοχή της εφαρμογής. Εκεί παρέχονται μερικές βοηθητικές λειτουργίες, όπως η σύνταξη και η αποθήκευση σημειώσεων για την επιλεγμένη συσκευή, καθώς και η προβολή του ιστορικού του nmap (βλ. και σχετικά άρθρα στο τεύχος 035) και των ευρημάτων που αφορούν στο συγκεκριμένο μηχάνημα. Στην ίδια περιοχή, όμως, παρέχονται και μερικές άκρως ενδιαφέρουσες επιλογές. Κατ' αρχάς, μπορούμε να συνδεθούμε σε κάποιο από τα ανοιχτά ports ή και να πραγματοποιήσουμε επιθέσεις brute force στις υπηρεσίες που προσφέρει η συσκευή-στόχος. Αλλά η σημαντικότερη επιλογή αφορά στην πραγματοποίηση επιθέσεων MiTM.



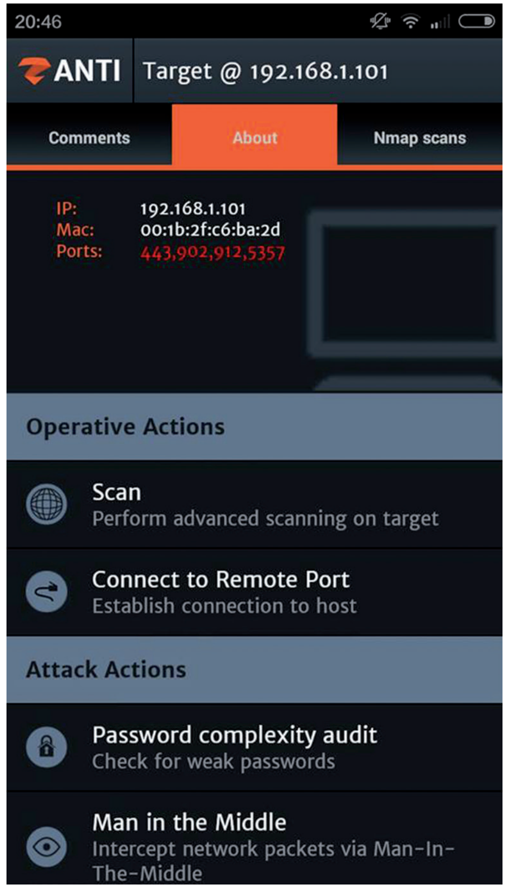
Η κεντρική "σελίδα" του zAnti 2. Εδώ βλέπουμε τα ευρήματα της αυτοματοποιημένης σάρωσης του δικτύου (network scanning). Η σχετική λίστα περιλαμβάνει τα μηχανήματα που βρέθηκαν, τις διευθύνσεις IP και MAC του καθενός, το λειτουργικό σύστημα και το πλήθος των ανοιχτών ports.

Αν ξεκινήσουμε μια επίθεση Man-in-The-Middle, το πρόγραμμα θα παρουσιάσει όλα τα δικτυακά sessions που δημιουργούνται με το θύμα, ενώ θα αρχίσει και την καταγραφή τους. Στην περίπτωση που σε κάποιο session εντοπισθεί η παρουσία ενός ζεύγους username-password, το πρόγραμμα εμφανίζει αυτόματα τα σχετικά strings. Από την άλλη, αν επιλέξουμε κάποια ιστοσελίδα με την οποία έχει συνδεθεί το σύστημα-στόχος, εμφανίζονται αρκετές νέες επιλογές. Μπορούμε να δούμε όλες τις συνδέσεις προς τον σχετικό ιστότοπο, ενώ επιχειρείται και ένα session hijacking μέσω του κλεμμένου cookie. Επιπρόσθετα, εμφανίζονται όλα τα στοιχεία σύνδεσης που είναι διαθέσιμα σε plain text, καθώς και οι πιθανές ευπάθειες του browser του θύματος. Παρεμπιπτόντως, στην περίπτωση που είμαστε εταιρικοί πελάτες της Zimperium μπορούμε να εκμεταλλευτούμε τις προαναφερθείσες ευπάθειες με τη βοήθεια της υπηρεσίας Zetasploit. Εκτός των άλλων, αν επιλέξουμε να αλληλεπιδράσουμε με κάποια ιστοσελίδα που επισκέπτεται το θύμα, εμφανίζονται όλες οι εικόνες που προέρχονται από τον ίδιο ιστότοπο και έχουν αποθηκευτεί στον browser του θύματος. Όπως καταλαβαίνετε, όταν πρόκειται για επιθέσεις MiTM το zAnti 2 δεν αστειεύεται καθόλου! Πέρα από όσα είπαμε, το ενσωματωμένο εργαλείο zPacketEditor επιτρέπει να τροποποιούμε σε πραγματικό χρόνο κάθε πακέτο HTTP που διέρχεται από τη συσκευή μας. Με αυτόν τον τρόπο μπορούμε να ανακατευθύνουμε τα αιτήματα HTTP του θύματος σε αυθαίρετες διευθύνσεις, να αλλάζουμε τις εικόνες που προσπαθεί να δει και να αντικαθιστούμε τα αρχεία που κατεβάζει. Τέλος, μπορούμε ακόμα και να εισάγουμε τον δικό μας κώδικα, στον κώδικα των ιστοσελίδων που επισκέπτεται το θύμα.

Προσθήκες στο zAnti

Αν πιστεύετε ότι η παραπάνω περιγραφή εξάντλησε τις δυνατότητες του zAnti 2, είσαστε γελασμένοι. Η Zimperium συνηθίζει να βγάζει διάφορα πρόσθετα για τις εφαρμογές της και παρ' όλο που μιλάμε για ένα νέο προϊόν, έχει ήδη κυκλοφορήσει ένα plugin κι ετοιμάζεται το επόμενο. Το υπάρχον πρόσθετο αφορά στις επιθέσεις ShellShock (βλ. άρθρο στο τεύχος 037) κι αποτελεί έναν ανιχνευτή γι' αυτή τη

διάσημη αλλά και ύπουλη ευπάθεια. Για την εγκατάσταση του plugin πρέπει να κατεβάσουμε και να εγκαταστήσουμε το ακόλουθο πακέτο: https://s3.amazonaws.com/zANTI/zANTI2ShellshockPlugin.apk. Πάντως, η ίδια λειτουργικότητα προσφέρεται και σε μια μικρή αυτόνομη εφαρμογή, που διατίθεται από την εταιρεία μέσω του Google Play. Για τους επόμενους μήνες η Zimperium φαίνεται να ετοιμάζει κάτι μεγάλο. Στο επίσημο blog της εταιρίας αναφέρεται η ανακάλυψη ενός νέου τύπου επιθέσεων MiTM, που ονομάστηκε DoubleDirect. Η συγκεκριμένη επίθεση επιτυγχάνει σε συσκευές που τρέχουν iOS, OS X και Android, ενώ αποτυγχάνει σε συσκευές Windows και Linux. Ευελπιστούμε ότι το σχετικό plugin για το zAnti 2 θα δημοσιευτεί σύντομα. Μέχρι τότε, σας αφήνουμε για να παίξουμε με το νέο μας παιχνίδι. Τα συστήματα ασφάλειας των μηχανημάτων μας θα στενάξουν ;)



Όταν επιλέγουμε μια από τις συσκευές που βρέθηκαν στο δίκτυό μας, μεταφερόμαστε σε μια άλλη περιοχή του προγράμματος. Από εδώ μπορούμε να πραγματοποιήσουμε διάφορες επιθέσεις, όπως επίσης και να επιχειρήσουμε τη σύνδεση σε κάποιο ανοιχτό port ή να πραγματοποιήσουμε μια προσαρμοσμένη ανίχνευση με το nmap. Πάντως το μεγαλύτερο ενδιαφέρον εστιάζεται στις επιθέσεις Man-in-The-Middle, όπου το πρόγραμμα παρέχει ένα τεράστιο σύνολο λειτουργιών.

ZIMPERIUM

Skill: Intermediate
Tags: iptables, ipset, BASH scripting, fail2ban



Ανελέητες επιθέσεις για καλύτερη άμυνα! [Μέρος 2/2]

Μπορεί οι επιθέσεις να αποκρούστηκαν αλλά ταυτόχρονα επιτεύχθηκε και ο στόχος τους. Σωστά διαβάσατε! Με το script που είδαμε στο άρθρο που αρχίζει από τη σελίδα 14, καταφέραμε να μπλοκάρουμε τις επιθέσεις και την ίδια στιγμή τις συνδέσεις των αθώων επισκεπτών :S Ευτυχώς, συνεχίζοντας την προσπάθειά μας καταφέραμε να εντοπίσουμε τα λάθη και να βρούμε μια ικανοποιητική λύση…

by Spir@l Evolut10n

Το σκριπτάκι μας για τη ρύθμιση του iptables έκρυβε ορισμένα λάθη. Για την ακρίβεια, έκρυβε την πρόχειρη και κατά κάποιον τρόπο επιφανειακή αντίληψη που είχαμε για τη λειτουργία του iptables. Ας δούμε όμως πώς ακριβώς συνεχίστηκαν οι δοκιμές μας. Όταν ολοκληρώσαμε το σκριπτάκι σπεύσαμε να το δοκιμάσουμε. Έτσι, το εκτελέσαμε για να "εγκαταστήσουμε" τους κανόνες μας στο iptables και ξεκινήσαμε μια επίθεση TCP flood με το LOIC. Στη συνέχεια προσπαθήσαμε να επισκεφθούμε το αντίγραφο του Parabing, που φιλοξενούσε το μηχάνημα–θύμα. Δεν ξέρουμε αν εσείς αντιληφθήκατε αμέσως το πρόβλημα, αλλά εμείς αιφνιδιαστήκαμε. Ο browser δεν κατάφερε να φορτώσει τη σελίδα ποτέ! Ας δούμε το γιατί, εξετάζοντας τις τελευταίες γραμμές του script:

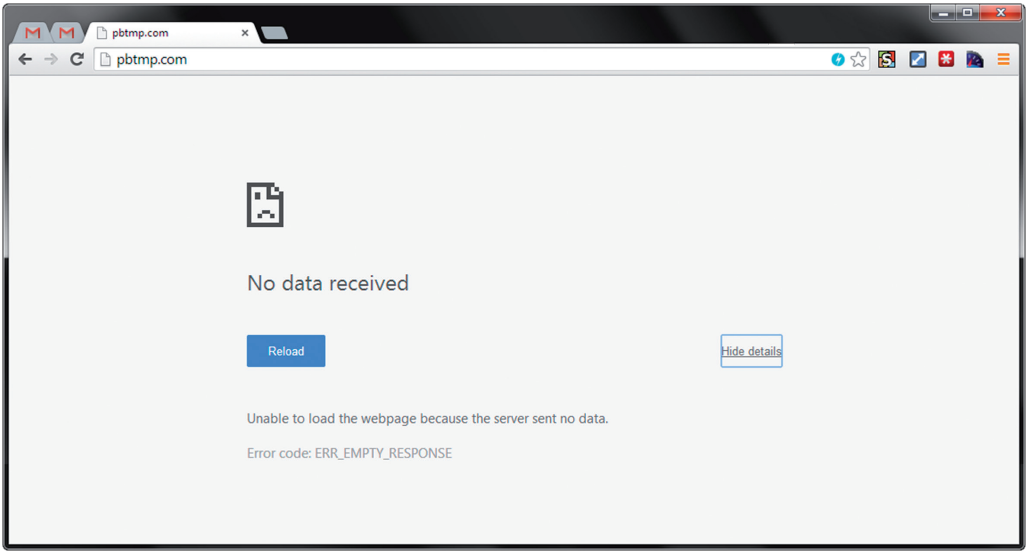
```
[...]
14 # block UDP flood
15 iptables -A inspect -p udp -m udp --dport 80 \
            -m limit --limit 1/sec --limit-burst 8 -j ACCEPT
16 iptables -A inspect -p udp -j log_n_drop
17 # block TCP PSH,ACK flood
18 iptables -A inspect -p tcp -m tcp --tcp-flags ALL PSH,ACK \
            -m limit --limit 1/sec --limit-burst 8 -j ACCEPT
19 iptables -A inspect -p tcp -m tcp --tcp-flags ALL PSH,ACK \
            -j log_n_drop
```

Το LOIC έστειλε τόσα πολλά πακέτα σε τόσο σύντομο χρονικό διάστημα, ώστε το όριο που είχαμε θέσει στη γραμμή 18 ξεπεράστηκε αμέσως. Έτσι, το iptables προχώρησε στον επόμενο κανόνα (γραμμή 19). Ο επόμενος κανόνας όμως, απορρίπτει *όλα* τα πακέτα που έχουν ενεργοποιημένα τα flags PSH και ACK, από *οπουδήποτε* κι αν προέρχονται. Με άλλα λόγια, δεν απορρίπτονται μόνο τα πακέτα που στέλνει το LOIC. Αυτή η βάρβαρη συμπεριφορά τραυμάτιζε και τελικά εμπόδιζε όλες τις επικοινωνίες μέσω TCP. Με τους κανόνες που είχαμε ορίσει, λοιπόν, τα πακέτα του LOIC απορρίπτονταν αλλά έπαιρναν μαζί τους και εκείνα των αθώων χρηστών. Στην περίπτωση του UDP συνέβαινε ακριβώς το ίδιο: Όταν ξεπερνιόταν το όριο που είχαμε θέσει, το iptables απέρριπτε όλα τα πακέτα UDP ανεξαιρέτως.

Ο τρόπος με τον οποίο είχαμε ρυθμίσει το iptables έκρυβε μία ακόμα παγίδα. Στην αλυσίδα inspect προωθούνταν τόσο τα εισερχόμενα στο σύστημα πακέτα, όσο και τα εξερχόμενα. Αυτό δεν αποτελούσε πρόβλημα από μόνο του. Όμως, οι κανόνες της αλυσίδας inspect δεν εξέταζαν το interface από το οποίο προέρχονταν τα πακέτα. Και κάπως έτσι προέκυπτε ένα μεγάλο πρόβλημα. Ας υποθέσουμε ότι δεχόμασταν μια επίθεση UDP flood στο port 80, με συνέπεια να ξεπεραστεί το όριο της γραμμής 15. Το iptables θα άρχιζε τότε να εφαρμόζει τον επόμενο κανόνα και τα πακέτα UDP θα προωθούνταν στην αλυσίδα log_n_drop. Αν εκείνη τη στιγμή προσπαθούσαμε να *στείλουμε* πακέτα UDP, τα πακέτα μας θα ακολουθούσαν τη μοίρα των εισερχόμενων και θα απορρίπτονταν...

Ποιος έφταιξε, τελικά;

Κατ' αρχάς πρέπει να τονίσουμε ότι η δομή του script στηριζόταν σε δύο ενδιαφέρουσες ιδέες: Σε αυτή της "ανακύκλωσης" ορισμένων κανόνων και σε εκείνη που



Δείτε τα αποτελέσματα από την πραγματοποίηση μιας επίθεσης TCP flood με το LOIC. Αν η επίθεση γινόταν στο Internet, από ένα μόνο μηχάνημα, η αποτελεσματικότητά της δεν θα ήταν δεδομένη. Στο τοπικό μας δίκτυο, όμως, ο web server γονάτισε αμέσως.

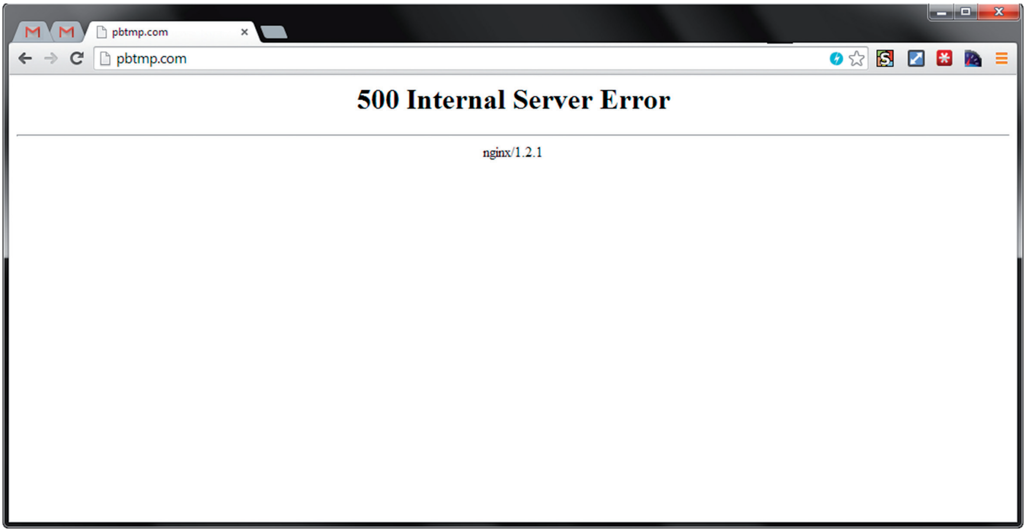
επιβάλλει τον έλεγχο και το φιλτράρισμα των εξερχόμενων πακέτων. Μπορεί η υλοποίησή μας να υπήρξε προβληματική, αλλά αυτές οι δύο ιδέες έχουν μεγάλη αξία. Η ανακύκλωση συνεπάγεται την ελαχιστοποίηση των κανόνων που θα γράψουμε και τους οποίους θα διαχειριζόμαστε στο μέλλον. Εν ολίγοις, τα οφέλη είναι παρόμοια με εκείνα της ανακύκλωσης του κώδικα γενικότερα. Επιπρόσθετα, είναι φρόνιμο να εξασφαλίζουμε ότι το μηχάνημά μας δεν πραγματοποιεί ποτέ επιθέσεις DoS, όχι μόνο για να μην μπλέκουμε χωρίς λόγο, αλλά και για να μην σπαταλάμε πολύτιμο bandwidth[1].

Μετά από τα παραπάνω ενδέχεται να προκύψει το εξής ερώτημα: Εφόσον αυτές οι δύο ιδέες είναι τόσο καλές και χρήσιμες, ποιος έφταιξε για την προβληματική συμπεριφορά του script; Η πικρή αλήθεια είναι ότι μοναδικός φταίχτης υπήρξε η προχειρότητα της σκέψης μας. Όταν γράφαμε τους κανόνες είχαμε εστιάσει τόσο πολύ στα πακέτα του LOIC που ξεχάσαμε τα υπόλοιπα, δηλαδή όσα προέρχονται από τους αθώους χρήστες. Η επιβολή ενός ορίου για το ρυθμό άφιξης των πακέτων δεν ήταν λανθασμένη ιδέα. Ωστόσο, αυτό το όριο έπρεπε να τεθεί για κάθε πιθανό επισκέπτη ξεχωριστά. Ξεπέρασες το όριο, κύριε; Κόβεσαι! Μόνο εσύ και όχι όλοι.

Διαχείριση συνόλων

Όταν εντοπίσαμε το λάθος αρχίσαμε να ανησυχούμε. Πώς ακριβώς θα διορθώναμε αυτό το πρόβλημα; Μήπως έπρεπε να γράψουμε έναν ξεχωριστό κανόνα για κάθε πιθανό επισκέπτη και συνεπώς για κάθε πιθανή διεύθυνση προέλευσης; Η σκέψη του να εισάγουμε τόσους πολλούς κανόνες στο iptables ήταν παρανοϊκή – δεν ήταν απλά παράλογη. Αυτή η δυσκολία γέννησε μια νέα απορία: Τι πρέπει να κάνουμε όταν θέλουμε να εφαρμόσουμε έναν κανόνα σε ένα σύνολο διευθύνσεων; Ακόμα κι αν το σύνολο περιέχει λίγες διευθύνσεις, η ιδέα του να γράψουμε πολλές φορές τον ίδιο

1. Ειδικά στους servers, το upstream bandwidth είναι πολύ πιο ακριβό από το downstream.



Η επίθεση HTTP flood από το LOIC έχει ελαφρώς διαφορετικά αποτελέσματα. Σε κάθε περίπτωση πάντως, ο web server πέφτει για πλάκα!

κανόνα είναι προβληματική. Καταρχάς, αν αποφασίσουμε αργότερα να αλλάξουμε τον κανόνα θα πρέπει να τον αλλάξουμε πολλές φορές και όχι μόνο μία. Εξάλλου, όσο αυξάνεται το πλήθος των κανόνων, τόσο επιβαρύνεται η λειτουργία του iptables.

Η αναζήτηση λύσης στο παραπάνω πρόβλημα μάς οδήγησε στο ipset. Πρόκειται για ένα εργαλείο που επιτρέπει τη δημιουργία και τη διαχείριση λιστών με διευθύνσεις IP, δικτυακές θύρες ή διευθύνσεις MAC. Αναρωτιέστε σε τι εξυπηρετούν αυτές οι λίστες; Ένα τμήμα της λειτουργικότητας του ipset προέρχεται από κώδικα που βρίσκεται στον πυρήνα του Linux και, όπως υποψιάζεστε, ο σχετικός κώδικας συνιστά επέκταση του iptables. Με τη βοήθεια του ipset μπορούμε να φτιάξουμε μία λίστα διευθύνσεων και στη συνέχεια να εφαρμόσουμε έναν κανόνα του iptables στα μέλη ολόκληρης της λίστας. Με αυτόν τον τρόπο θα έχουμε εισαγάγει μόνο μια φορά τον κανόνα και η επιβάρυνση στο iptables θα είναι η ελάχιστη δυνατή. Επιπρόσθετα, θα μπορούμε να προσθαφαιρούμε εγγραφές στη λίστα και κατά συνέπεια να επεκτείνουμε ή να περιορίζουμε την εφαρμογή του κανόνα, με μεγάλη ευκολία.

Ωραία όλα αυτά, θα πείτε, αλλά όταν εφαρμόζουμε έναν κανόνα σε μία λίστα, δεν εφαρμόζεται *ξεχωριστά* για κάθε εγγραφή της λίστας. Με άλλα λόγια, αν χρησιμοποιούσαμε λίστες του ipset στις γραμμές 15 έως 19 του προηγούμενου script, θα αντιμετωπίζαμε πάλι το ίδιο πρόβλημα. Το iptables δεν θα ήξερε ποιο από τα μέλη της λίστας παραβίασε το όριο και θα τα τιμωρούσε όλα. Πράγματι! Το ipset δεν μπορεί να λύσει όλα τα προβλήματά μας.

Νέα προσέγγιση

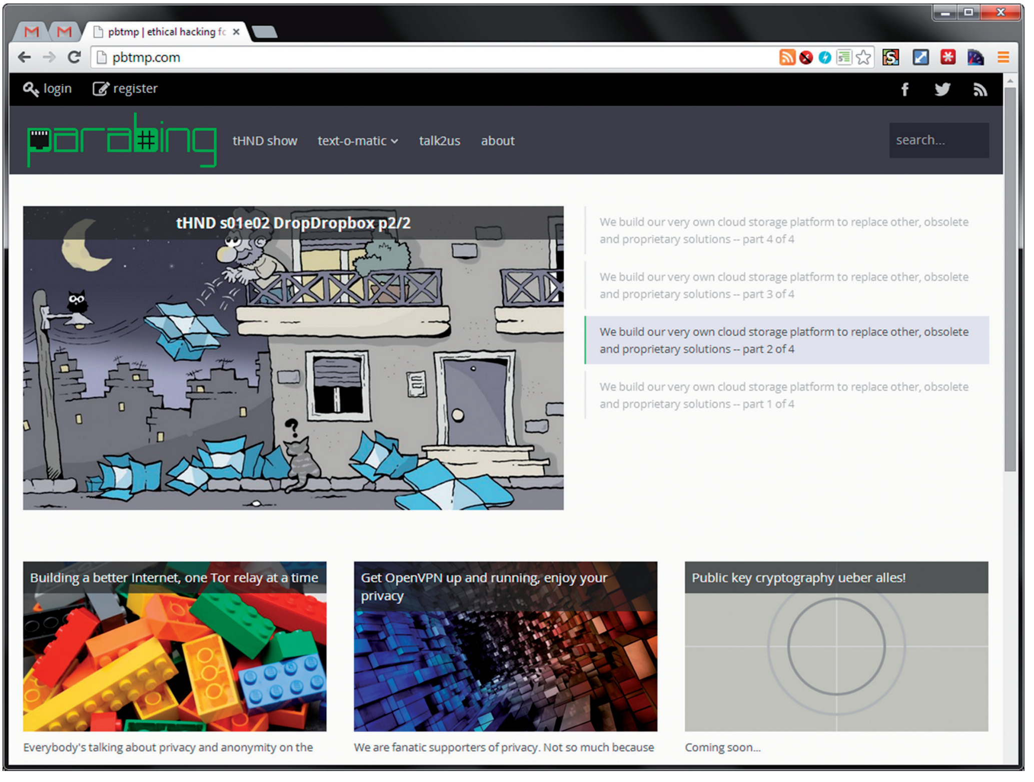
Η λειτουργία του ipset φαινόταν πολύ ενδιαφέρουσα και, σε τελική ανάλυση, η αξιοποίησή του θα αποτελούσε μια καλή ευκαιρία για το γνωρίσουμε καλύτερα. Έτσι, σκεφτήκαμε να δημιουργήσουμε μια λίστα για τους αθώους επισκέπτες ή μία για τους κακόβουλους. Στην πρώτη περίπτωση θα δημιουργούσαμε κι έναν κανόνα που θα επέτρεπε τη σύνδεση στα μέλη της λίστας, ενώ στη δεύτερη θα φτιάχναμε έναν κανόνα που θα μπλόκαρε κάθε επικοινωνία. Με αυτόν τον τρόπο θα μπορούσαμε να

αντιμετωπίσουμε κάθε χρήστη με το σωστό τρόπο, χωρίς να εισάγουμε ένα σωρό κανόνες στο iptables. Η σκέψη και μόνο μιας τόσο αποδοτικής λύσης μάς ικανοποιούσε πάρα πολύ, αλλά δημιουργούσε ένα νέο πρόβλημα: Ποιος θα φρόντιζε για την ενημέρωση της λίστας του ipset; Αρχικά σκεφτήκαμε να αναθέσουμε τη δουλειά στο ίδιο το iptables. Βλέπετε, το ipset δεν είναι καινούργιο φρούτο και το iptables ενσωματώνει αρκετούς μηχανισμούς για την προσθήκη διευθύνσεων σε λίστες του ipset. Για του λόγου το αληθές, δείτε το ακόλουθο παράδειγμα. Ας υποθέσουμε ότι έχουμε φτιάξει μια λίστα για τους κακόβουλους χρήστες. Κάτι τέτοιο μπορεί να γίνει πανεύκολα δίνοντας το εξής:

```
ipset -N banned iphash
```

Η λίστα που θα δημιουργηθεί με αυτόν τον τρόπο θα έχει το όνομα banned και θα φιλοξενεί διευθύνσεις IP. Ας υποθέσουμε επίσης ότι θέλουμε να μπλοκάρουμε οποιονδήποτε προσπαθεί να συνδεθεί στο port 22, επειδή το μηχάνημά μας δεν παρέχει καμία υπηρεσία σε αυτή τη θύρα και θεωρούμε ότι ο χρήστης συμπεριφέρεται ύποπτα. Αυτό θα μπορούσε να γίνει πανεύκολα με την προσθήκη των εξής κανόνων:

```
iptables -A INPUT -p tcp --dport 22 -j SET --add-set banned src
iptables -A INPUT -m set --match-set banned src –j DROP
```



Κατά πάσα πιθανότητα δεν θα μπορέσετε να νιώσετε την αγαλλίασή μας! Πρόκειται για αντίγραφο του Parabing που σερβίρει ο web server του τοπικού μας δικτύου. Μετά από τη ρύθμιση του iptables και την εκτέλεση του script που ενημερώνει αυτόματα τη λίστα banned, οι επιθέσεις του LOIC έπεφταν στο κενό!

Ο πρώτος κανόνας προσθέτει στη λίστα banned τη διεύθυνση IP οποιουδήποτε επιχειρεί να συνδεθεί στο port 22 μέσω TCP. Από την άλλη, ο δεύτερος κανόνας απορρίπτει όλα τα πακέτα που προέρχονται από κάποια διεύθυνση της λίστας banned. Αυτή ήταν όλο! Οποιοσδήποτε προσπαθεί να συνδεθεί στο port 22 θα προστίθεται αυτόματα στη λίστα banned και στη συνέχεια τα πακέτα του θα απορρίπτονται. Το κολπάκι που παρουσιάσαμε έχει αρκετό ενδιαφέρον, αν και στην πραγματικότητα θα αποτελούσε ένα πολύ αυστηρό μέτρο. Ας επιστρέψουμε όμως στο θέμα μας…

Η ενημέρωση της λίστας από το ίδιο το iptables ήταν εφικτή, αλλά πολύ σύντομα διαπιστώσαμε ότι οδηγούσε σε αδιέξοδο: Πώς θα διέκρινε το iptables τις κακόβουλες συνδέσεις; Με τη βοήθεια κάποιου ορίου; Και που θα εφαρμοζόταν ο σχετικός έλεγχος; Στον καθένα ξεχωριστά ή σε όλους μαζί; Αυτά τα ερωτήματα μάς βοήθησαν να καταλάβουμε ότι μετά από έναν τεράστιο κύκλο, είχαμε φτάσει πάλι στο αρχικό πρόβλημα. Κάπως έτσι, αποφασίσαμε να προχωρήσουμε σε πιο γενναία μέτρα για την αντιμετώπιση των επιθέσεων DoS. Διατηρήσαμε την ιδέα της λίστας για τους κακόβουλους χρήστες, αλλά για την ενημέρωσή της αποφασίσαμε να επιστρατεύσουμε το κέλυφος BASH. Ο εντοπισμός των συνδεδεμένων χρηστών πραγματοποιείται πανεύκολα από το κέλυφος και το ίδιο ισχύει για την καταμέτρηση των συνδέσεων του καθενός. Επομένως, θα μπορούσαμε να αναθέσουμε τους σχετικούς ελέγχους σε ένα BASH script, ώστε το iptables να αναλάβει αποκλειστικά τα πιθανά μπλοκαρίσματα.

Περισσότερος κώδικας

Αρχικά, ανοίξαμε το script για τη ρύθμιση του iptables και πραγματοποιήσαμε τις απαραίτητες προσθήκες. Δείτε την τελική του μορφή:

```
iptables -F
iptables –X
ipset –F
ipset –X
ipset –N banned iphash
iptables -N inspect
iptables -N log_n_drop
iptables -P INPUT DROP
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth0 -m set --match-set banned src -j log_n_drop
iptables -A INPUT -i eth0 –p tcp –dport 80 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -P OUTPUT ACCEPT
iptables -A OUTPUT -m state --state INVALID -j DROP
iptables -A OUTPUT -j inspect
iptables -A log_n_drop -j LOG --log-prefix '[suspected DoS] '
iptables -A log_n_drop -j DROP
# block UDP flood
```

```
19 iptables -A inspect -p udp -m udp --dport 80 \
            -m limit --limit 1/sec --limit-burst 8 -j ACCEPT
20 iptables -A inspect -p udp -j log_n_drop
21 # block TCP PSH,ACK flood
22 iptables -A inspect -p tcp -m tcp --tcp-flags ALL PSH,ACK \
            -m limit --limit 1/sec --limit-burst 8 -j ACCEPT
23 iptables -A inspect -p tcp -m tcp --tcp-flags ALL PSH,ACK \
            -j log_n_drop
```

Οι νέες γραμμές σημειώνονται με τα έντονα γράμματα. Πριν αναφερθούμε σ' αυτές, όμως, αξίζει να σταθούμε για λίγο στην αλυσίδα inspect. Όπως βλέπετε, όχι μόνο δεν την καταργήσαμε, αλλά διατηρήσαμε και τους κανόνες της. Η συγκεκριμένη αλυσίδα αξιοποιείται πλέον μόνο για τα εξερχόμενα πακέτα. Επομένως, δεν υπάρχει ο κίνδυνος να τιμωρηθεί κάποιος χρήστης κατά λάθος. Σε έλεγχο υποβάλλονται μόνο οι εξερχόμενες συνδέσεις του μηχανήματος και αν παραβιαστούν τα όρια που έχουμε θέσει θα διακοπούν μόνον αυτές. Το ipset αξιοποιείται αποκλειστικά για το μπλοκάρισμα των κακόβουλων χρηστών. Ο σχετικός κανόνας εμφανίζεται στη γραμμή 10: Όσες συνδέσεις προέρχονται από διεύθυνση που ανήκει στο σύνολο banned, προωθούνται στην αλυσίδα log_n_drop. Φυσικά, πριν από οποιαδήποτε αναφορά στην αλυσίδα banned, απαιτείται η κατάλληλη "αρχικοποίηση" του ipset. Έτσι, στις γραμμές 03 και 04 διαγράφουμε τα περιεχόμενα των συνόλων αλλά και τα ίδια τα σύνολα που ενδέχεται να προϋπήρχαν, ενώ στη γραμμή 05 δημιουργούμε το σύνολο banned. Μια νέα προσθήκη στο script αποτελεί και η γραμμή 11. Η παρουσία της μάλιστα είναι κρίσιμη! Θυμηθείτε ότι σαν γενική πολιτική (policy) της αλυσίδας INPUT έχουμε επιλέξει το DROP. Αυτό σημαίνει ότι πρέπει να δημιουργήσουμε έναν κανόνα που θα *επιτρέπει* τη σύνδεση στο port 80 από το internet (eth0). Χωρίς αυτόν, τα σχετικά πακέτα θα απορρίπτονται εξ ορισμού κι ανεξάρτητα από το αν προέρχονται από διευθύνσεις του συνόλου banned ή όχι.

Για τη συγκρότηση του μηχανισμού που είχαμε σκαρφιστεί, γράψαμε και το ακόλουθο script:

```
#!/bin/bash
while true ; do
 netstat -ntu | \
    grep 'ESTABLISHED' | \
    tail -n +3 | \
    awk '{print $5}' | \
    cut -d: -f1 | \
    sort -n | \
    uniq -c > /var/log/.tmp_ip_list
 while read line; do
    cn=$(echo $line | cut -d" " -f1)
    ip=$(echo $line | cut -d" " -f2)
    if [ "$cn" -gt "8" ]; then
```

```
07          ipset -A banned $ip > /dev/null 2>&1
08      fi
09    done < /var/log/.tmp_ip_list
10    sleep 30
11 done
```

Η διάρθρωση του κώδικα είναι αρκετά απλή. Καταρχάς ο "εξωτερικός" βρόχος while (γραμμές 02 έως 11) επαναλαμβάνεται για πάντα, με μια καθυστέρηση των 30 δευτερολέπτων (γραμμή 10). Έτσι, ανά μισό λεπτό πραγματοποιείται ένας νέος έλεγχος, για τον εντοπισμό των χρηστών που επιτίθενται στο σύστημα. Ο σχετικός έλεγχος ξεκινά με τη δημιουργία μιας ειδικής λίστας. Σε κάθε γραμμή αυτής της λίστας εμφανίζονται οι διευθύνσεις IP των συνδεδεμένων χρηστών, ενώ δίπλα σε κάθε διεύθυνση εμφανίζεται το πλήθος των συνδέσεων TCP και UDP (σαν σύνολο, αλλά αυτό δεν μας ενοχλεί). Όλη η δουλειά γίνεται στην τεράστια γραμμή 02. Όπως βλέπετε, ξεκινάμε με το netstat για να πάρουμε μια πρώτη λίστα με τους συνδεδεμένους χρήστες και στη συνέχεια, με τη βοήθεια της διασωλήνωσης και μερικών εργαλείων της γραμμής εντολών, σχηματίζουμε την τελική λίστα. Στο τέλος, με τη βοήθεια της ανακατεύθυνσης, η λίστα αποθηκεύεται σε ένα προσωρινό αρχείο (/var/log/.tmp_ip_list). Αμέσως μετά ακολουθεί ένας ακόμα βρόχος while, ο οποίος διαβάζει μία προς μία τις γραμμές του παραπάνω αρχείου. Για κάθε γραμμή, οι τιμές που αναφέρονται τοποθετούνται σε δύο κατάλληλες μεταβλητές (γραμμές 04 και 05). Στη γραμμή 06 πραγματοποιείται ο κρίσιμος έλεγχος: Μήπως οι ενεργές συνδέσεις από την τάδε διεύθυνση έχουν ξεπεράσει τις οκτώ; Αν ναι, η συγκεκριμένη διεύθυνση προστίθεται στη λίστα banned! Όπως αντιλαμβάνεστε, το όριο των οκτώ ενεργών συνδέσεων επιλέχθηκε σχεδόν αυθαίρετα. Στηριχτήκαμε μόνο στις παρατηρήσεις μας για το μέσο πλήθος των ενεργών συνδέσεων, που πραγματοποιούνται κατά την περιήγηση ενός τυπικού επισκέπτη.

Διπλή νίκη

Μετά από τη νέα μας προσπάθεια να προστατεύσουμε το σύστημα, ακολούθησε και μια νέα δοκιμή. Περιττό να σας πούμε ότι αυτή τη φορά τα είχαμε καταφέρει! Η διεύθυνση του εικονικού μηχανήματος από το οποίο εξαπολύονταν οι επιθέσεις, προστέθηκε αμέσως στη λίστα banned και το iptables διέκοψε κάθε επικοινωνία μαζί του. Ταυτόχρονα, εμείς μπορούσαμε να συνδεθούμε ανενόχλητοι και να περιηγηθούμε ξέγνοιαστα στο αντίγραφο του Parabing ;) Εννοείται πως η χαρά μας ήταν απερίγραπτη, μέχρι που εμφανίστηκε ένας μικρός προβληματισμός. Είχαμε αναλωθεί τόσο πολύ στην απόκρουση των επιθέσεων TCP και UDP flood, που είχαμε ξεχάσει εντελώς τις επιθέσεις HTTP flood. Για την αντιμετώπιση αυτών των επιθέσεων είχαμε σκεφτεί να επιστρατεύσουμε το fail2ban. Μήπως έπρεπε να αφήσουμε τις χαρές και τα πανηγύρια, για να στρωθούμε πάλι στη δουλειά; Ευτυχώς όχι! Ο μηχανισμός που είχαμε φτιάξει ήταν αποτελεσματικός απέναντι *και* σε αυτές τις επιθέσεις :D

Έχετε υπόψη ότι μπορείτε να κατεβάσετε το script για τη ρύθμιση του iptables από το `http://bit.ly/dh039iptconf` και το script για την αυτόματη ενημέρωση της λίστας banned από το `http://bit.ly/dh039ipsetupd`. Θα σας αφήσουμε να μελετήσετε τα σκριπτάκια μας με την ησυχία σας και, ποιος ξέρει, ίσως να βρείτε ατέλειες και προβλήματα που μας διέφυγαν. Μη νομίζετε ότι κάτι τέτοιο θα μας δυσαρεστήσει. Ίσα-ίσα που θα μας δώσετε την ευκαιρία να μελετήσουμε παραπάνω. Ελπίζουμε να κάνετε κι εσείς το ίδιο!

COOKIES


ΦΗ
ΩΝ
ΛΑ
...ΦΣΣΣ...
ΦΗ
ΤΩΝ
ΜΕΝΩ
ΥΛΑ
ΟΚ
...
CAM 1
ΜΕ ΑΥΤΟ ΤΟ ΚΕΙΜΕΝΟ
ΣΤΟΝ ΤΟΙΧΟ ΜΑΣ,
ΔΗΛΩΝΟΥΜΕ ΠΩΣ ΔΕΝ
ΣΥΝΑΙΝΟΥΜΕ ΣΤΗΝ ΚΑΤΑΓΡΑΦΗ,
ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΧΡΗΣΗ ΤΩΝ
ΠΡΟΣΩΠΙΚΩΝ ΜΑΣ ΔΕΔΟΜΕΝΩΝ
ΑΝΕΣΤΗΣ-ΧΡΥΣΟΥΛΑ
ΠaNOS
...ΝΟΜΙΚΑ ΤΟΥΛΑΧΙΣΤΟΝ ΕΙΜΑΣΤΕ ΚΑΤΟΧΥΡΩΜΕΝΟΙ...

deltahacker.gr